A Reproduced Copy

OF N71-19791

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

RECONFIGURABLE G & C COMPUTER STUDY FOR SPACE STATION USE

FINAL REPORT

VOLUME IV

APPENDIX 2

31 January 1971

J. Jurison
Program Manager

Prepared Under Contract No. NAS9-10416

Autonetics Division of North American Rockwell
for
Manned Spacecraft Center
National Aeronautics and Space Administration

FOREWORD

This final report covers the work performed by Autonetics Division of North American Rockwell Corporation under a study contract entitled Reconfigurable G&C computer Study for Space Station Use. The report is submitted to the National Aeronautics and Space Administration Manned Spacecraft Center under the requirements of Contract NAS 9-10416. The study program covered the period from December 29, 1969 through January 31, 1971. The NASA Technical Monitor was Mr. E. S. Chevers.

The final report consists of seven (7) volumes:

Volume I	Technical Summary
Volume II	Final Technical Report
Volume III	Appendix I. Model Specification
➧Volume IV	Appendix 2. IOP - VCS Detailed Design
Volume V	Appendix 3. System Analysis and Trade-Offs
Volume VI	Appendix 4. Software and Simulation Description and Results
Volume VII	Appendix 5. D-200 Computer Family
	Appendix 6. System Error Analysis
	Appendix 7. Reliability Derivation for Candidate Computers
	Appendix 8. Power Converter Design Data
	Appendix 9. Data Transmission Medium Design

➧YOU ARE READING THIS VOLUME

VOLUME IV
APPENDIX 4
TABLE OF CONTENTS

VOLUME IV
APPENDIX 2
LIST OF ILLUSTRATIONS

VOLUME IV
APPENDIX 2
LIST OF TABLES

1. IOP/VCS Operation

1.1 System Concept

The computer system consists of four computers and four I/O buses that interface with local processors of external subsystems. These computers are interconnected for special I/O functions to achieve a Fail Op-Fail Op-Fail Safe (FOOS) computer system as shown in Figure 1. Each computer contains a connection to one of the four external I/O buses in the system. In addition, each computer contains four interconnections: three receive channels, one from each of the other computers, and one output channel to all the other computers; it is via these connections that the voter-comparator-switch (VCS) concept is mechanized.

The VCS function is deeply interrelated to the IOP in the computer. However, it is depicted as a unique module in Figure 2 where the interconnection is shown in greater detail. The VCS concept enables the computer system to be operated in a variety of modes: Four-way voting (all four computers doing the same job, with one or more of the VCS' voting on the I/O information), three-way voting with the other computer doing a different job, two-way comparison with the other two computers also in comparison or doing distinct jobs, and four non-redundant computers. The mode is under the control of the executive system which is distributed among all the computers. The executive is redundant, in a distributed sense, to satisfy the FOOS requirement. The level of redundancy of the computations may be changed under software control by changing the mode of the system.

The four computers each operate on its own independent clock. Redundant data operated on by the VCS is required to be in synchronization within a specified tolerance; this data is not required to be in bit sync. The computer system, by a combination of hardware and software, is self-reconfigurable and is capable of withstanding any three failures in a FOOS manner.

1.2 VCS Architecture

The VCS device is capable of operating on redundant data in a majority voting or a comparison mode, thereby performing a redundancy reduction of either 4:1, 3:1, or 2:1, or it may operate on non-redundant data. The device is adaptive in that it may be switched into different modes. The device is also adaptive to failures in the computer system by means of adaptive majority logic.

The diagram indicating the basic architecture of the VCS device and the interface between the VCS and IOP is shown in Figure 3. The VCS device outputs on the I/O bus to the external subsystems; it has as inputs, the outputs of the four I/O processor sections of the four computers. As shown, these inputs to the VCS may be used by the voting, comparison, or selector logic. The block containing this logic is directed by the control unit of the VCS.

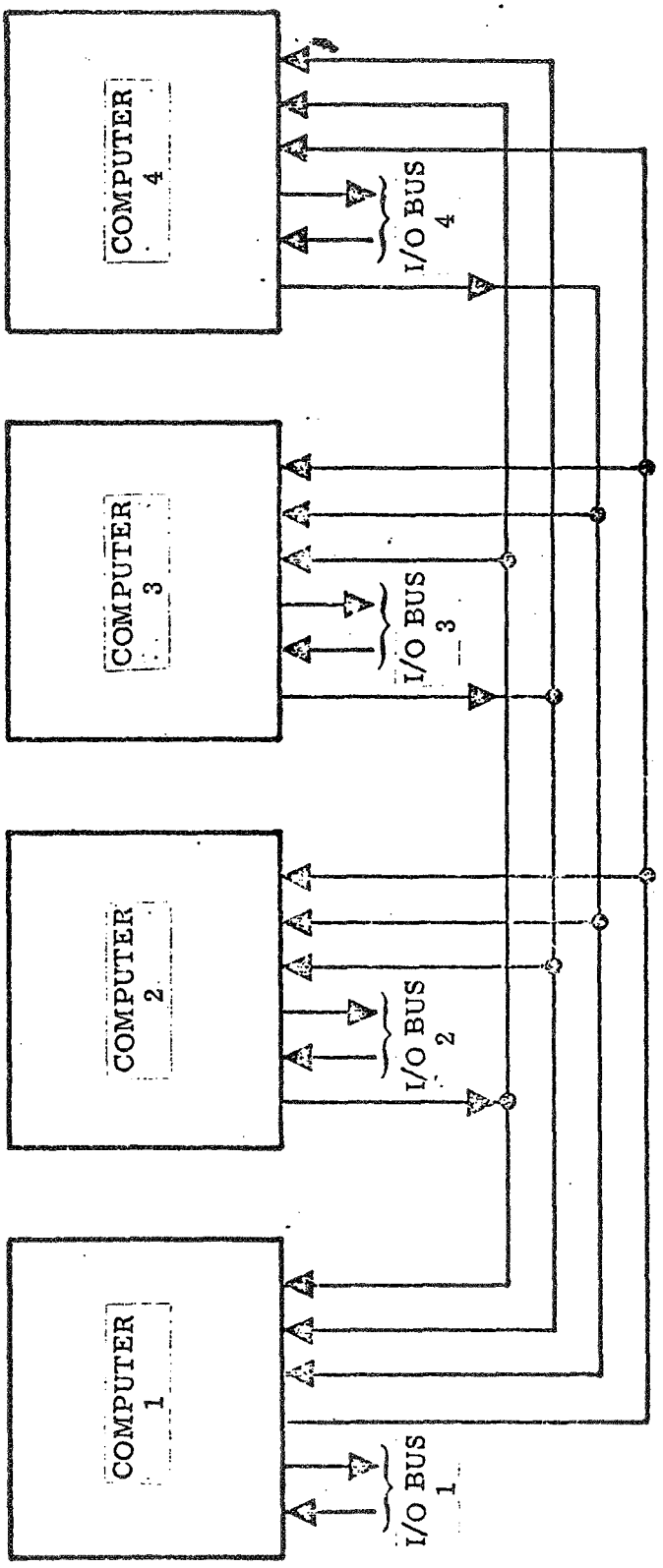


Figure 1-1. COMPUTER SYSTEM INTERCONNECTION DIAGRAM

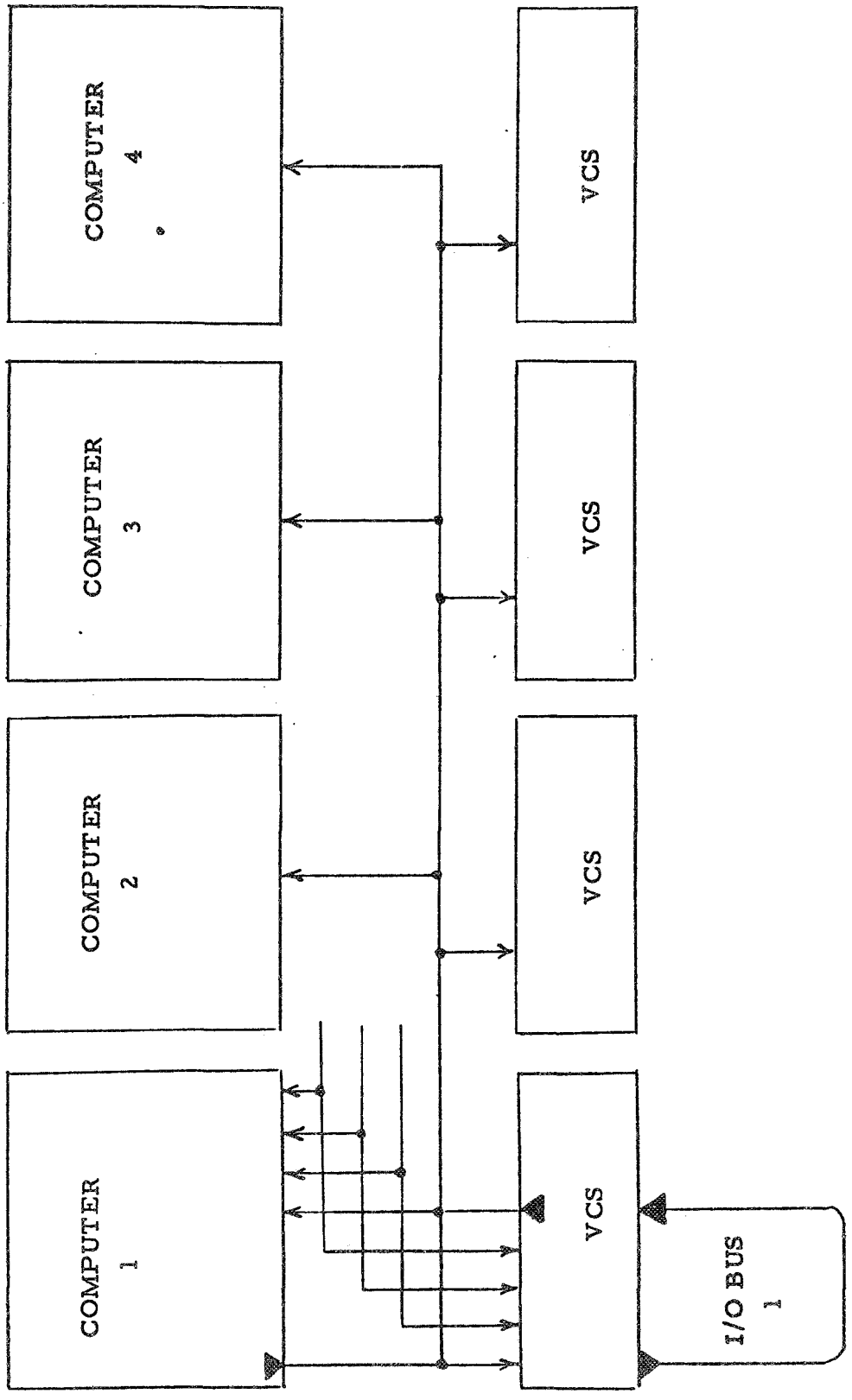


FIGURE 1-2. COMPUTER SYSTEM INTERCONNECTION MECHANIZATION

The control unit operates on the principle of adaptive majority logic. Its function is to control the mode of the voter comparator selector logic block and to decide which computers are failed or non-failed. The control unit implements its functions by means of a P matrix and a R matrix. The P matrix (4 x 4) contains one computer's failure status opinion of another computer and the majority decision of the failure status of each computer. Essentially this matrix contains the failure status of the computer system. The failure status decision on an individual computer is derived from the other computers' failure opinion and the go-no go self test results from the built-in test equipment in each computer. The decisions are arrived at on the basis of adaptive majority logic.

The R matrix of the control unit represents the desired mode of operation (four-way voter, three-way voter, two-way comparator, or selector). It operates under a majority decision rule as to the selection of the mode. Further, it is adaptive in that the P matrix is used to determine which computers are failed and should be ignored in the R matrix. The P and R matrices are set by commands received from the I/O processors.

Since the four computers are not operated in bit sync when in the voting or comparison modes, a set of buffer shift registers are provided in the VCS to provide bit synchronous data to the voter comparator selector logic. The registers allow for a ± 1/2 word time out of sync operation. The I/O bus to the external subsystems is a continuous loop providing the ability to monitor outputs of the computer on to the bus by echo check techniques of monitoring the inputs from the bus. Routing logic in the VCS device determines the destination of all input data received over the I/O bus by the VCS. This routing logic also determines which I/O processors are to receive copies of the input data from the external subsystems (e.g., in a three-way voting mode input data will be distributed to three I/O processors).

1.3 IOP Architecture

The Input/Output processor (IOP) functions as an independent processor operating under a stored program in memory and capable of interfacing with the internal memory directly via the memory bus. A block diagram of the basic functions of the IOP is given in Figure 4. The VCS function is included with the IOP in Figure 4 since it is so closely interrelated. The input/output functions may be classified into three different types:

A. Type 1 - for computer-to-computer communication

B. Type 2 - for computer-to-external subsystem communication

C. Type 3 - for computer-to-parallel channel communication

Type 1 channel communications are bit serial-word serial. The channels are completely independent so that the IOP may be simultaneously receiving information from three other IOPs and sending information to these IOPs on its Type 1 output channel. The information sent over the Type 1 channels may contain data or commands. Further, the information may be destined for the IOP or the VCS. Likewise, the information sent out on the Type 1 channel may originate from the IOP or the VCS. Proper addresses

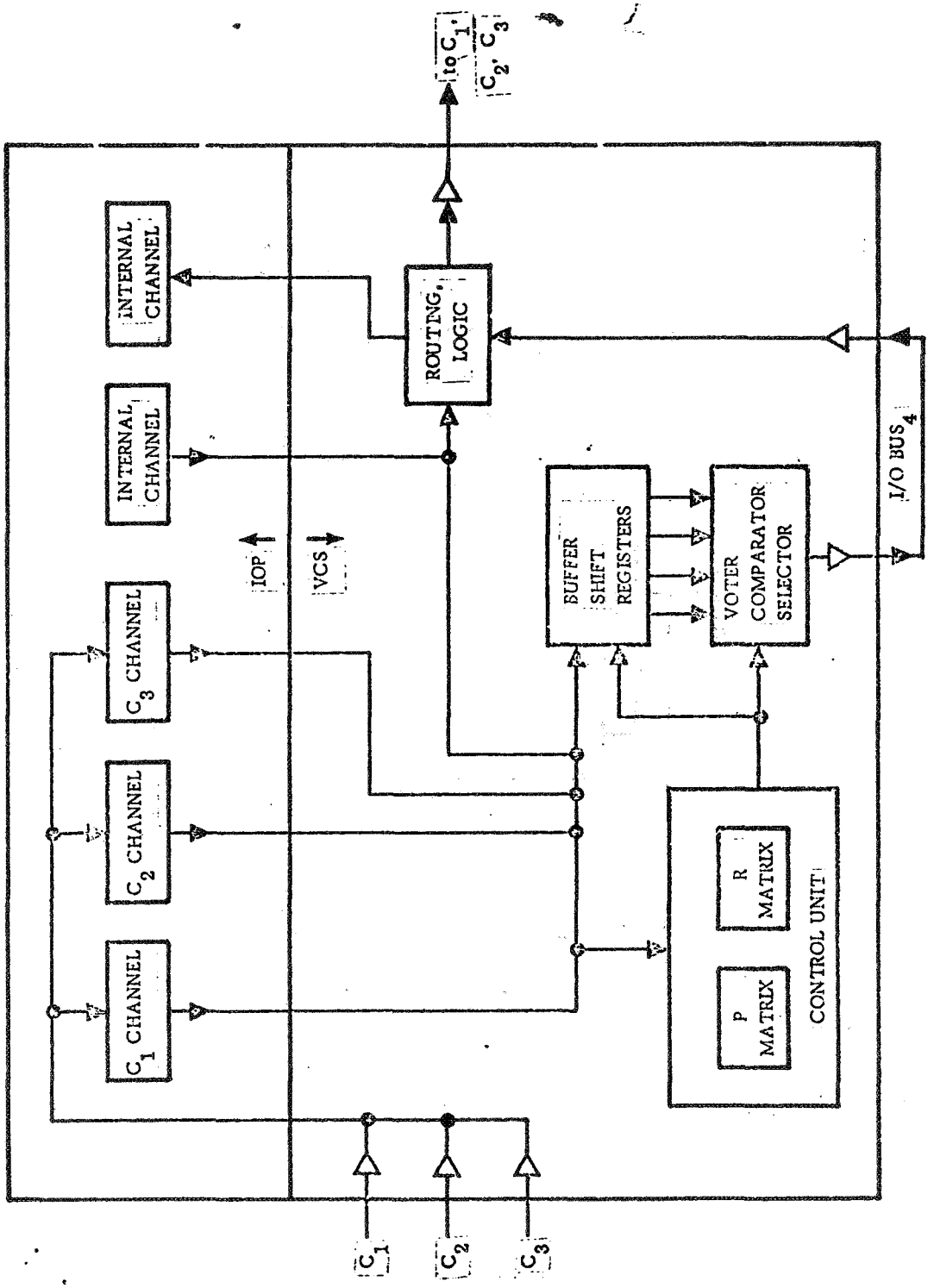


Figure 1-3. IOP - VCS BLOCK DIAGRAM

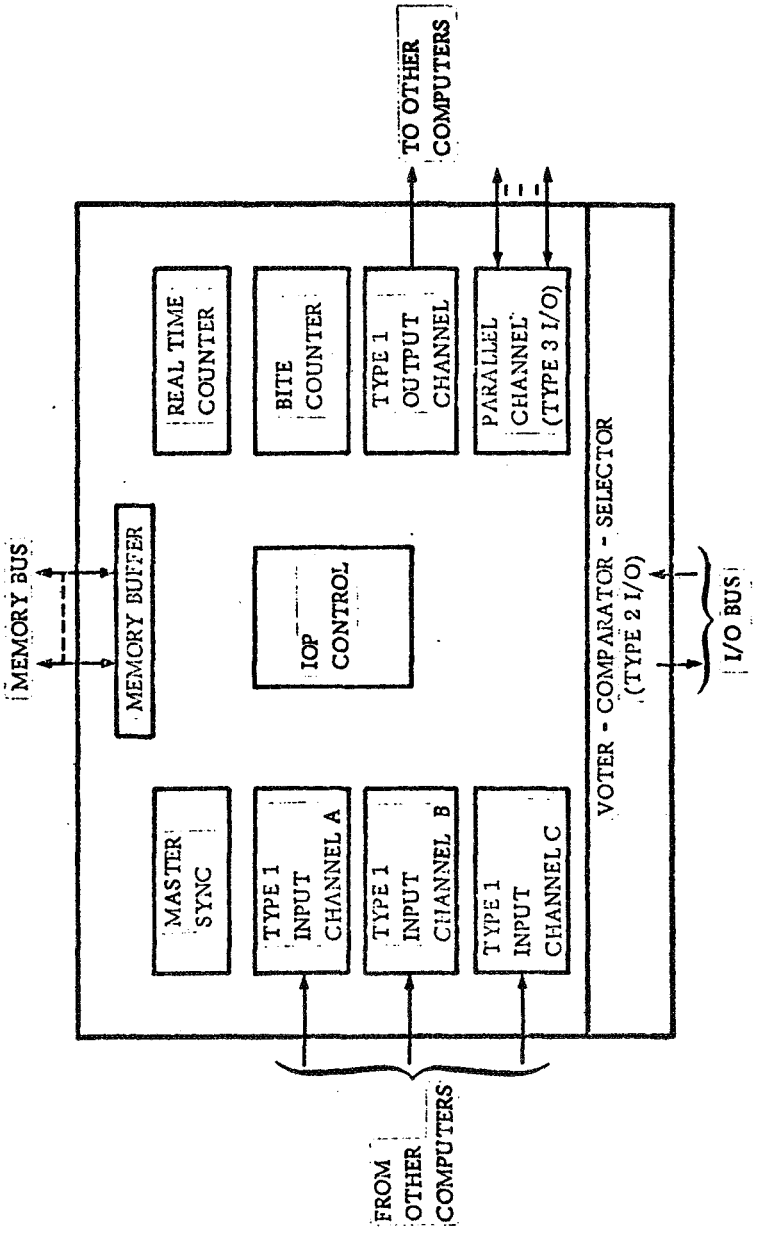


Figure 1-4. IOP DIAGRAM

accompany messages over the Type 1 channels that allow the information to be routed to the proper computer and the appropriate parts of the IOP and VCS.

Type 2 channel communications are also bit serial-word serial. The Type 2 channel interfaces with the I/O bus that connects to the various external subsystems. The use of the Type 2 channel is completely under control of the IOP; external subsystems may only communicate with the IOP and only after being commanded to do so by the IOP. The Type 2 channel is closely interrelated to the VCS function as described in Section 1.2.

In addition to these serial channels, the IOP contains a parallel I/O channel (16-bit) shown as the Type 3 channel. This channel is used for communication to devices requiring rapid data transfer (250,000 words/sec) with the computer system such as the mass storage system and the data management system. The channel operates under external control only, i.e., the IOP does not initiate data transfers over this channel. The Type 3 channel is not included in the FOOS requirement.

The IOP is set into operation by the decoding of a command and/or a control word. The commands are stored in the computer memory and are accessed according to the command program counter in the IOP. Control words are also stored in the memory and are also received from other computers in the system over the inter-computer (Type 1) bus. Commands are executed as in any other software program. Control words are executed only when specified by a command or when a control word is received over the inter-computer bus.

The IOP is capable of executing the following commands:

1. Halt and proceed
2. Jump
3. Conditional skip
4. Fetch control word

Figure 5 indicates a typical layout of the IOP program. The operation of the fetch control word can be seen to mechanize data transfer operations initiated by the IOP. Each fetch control word instruction contains an address that points to the location of a control word. The control word contains information that describes the type of operation to be performed (e.g., input from local processors, output to another computer, etc.). Following each control word is an address that points to the location of a data block. The data block is the location into which input/output data is to be placed or sent from.

Control words may also be received from another computer over the Type 1 I/O channels. The internal and external control words mechanize the complete set of IOP information transfer operations. These include not only data transfer between computers and between computers and external subsystems, but also between computers and the VCS devices.

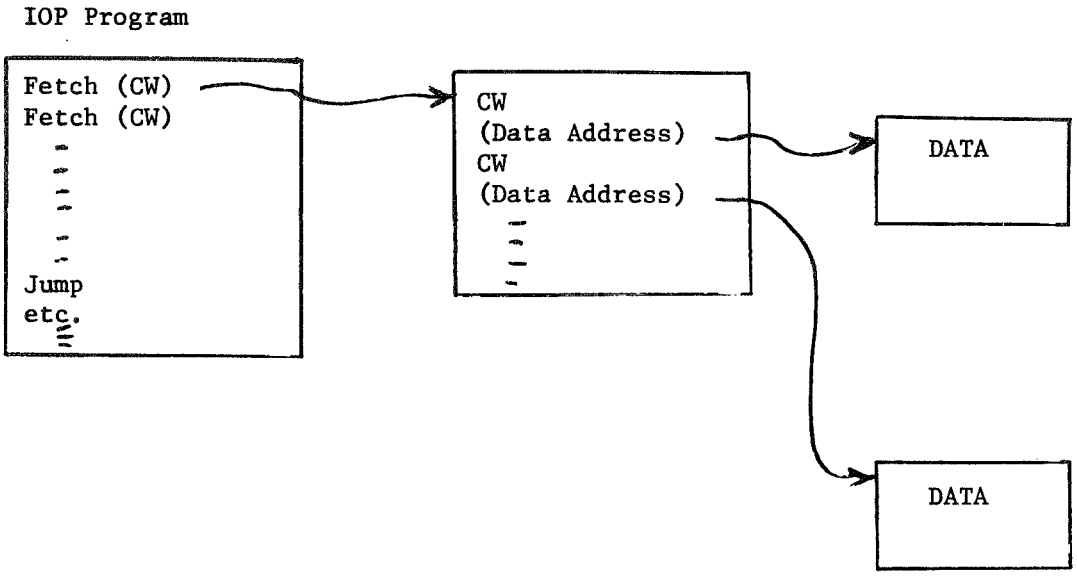


Figure 5. IOP Program Operation

A real time counter is included in the IOP to provide a repetitive time period reference for IOP and CPU program synchronization. The counter is counted down to zero from a set value at the logic clock rate. At zero, the counter issues an interrupt to the CPU and IOP mode controls, resets itself and starts counting down again. The length of the time period is reprogrammable.

The VCS device constitutes the primary means of failure detection in the computer system since an erroneous input to the voter may be readily detected. In addition, each computer contains self test capability that provides autonomous failure detection to a specified confidence level; this capability is used to aid the computer system failure reconfiguration operations. The self test is a combination of hardware/software techniques. The IOP contains some of this hardware in the form of the BITE Counter. This counter operates such that if it is not reset periodically, it will reach a zero count and issue a computer failure signal. The counter will be reset by accessing a dedicated location in memory and loading the binary value found therein into the counter and zeroing this location after access. It is the CPU's responsibility to reload this location to keep the counter reset.

1.4 IOP/System Operation

The computer system may be operated in a variety of redundancy modes. The modes are implemented by the hardware design features in the VCS. This hardware is set up and controlled by software executive control in the computer system. This executive is distributed among the four computers with the VCS hardware acting upon the software control by an adaptive majority rule. Hard core software failures are eliminated by this design approach. The VCS is designed such that the condition of its operating state and the results of voting/comparison processes may be interrogated by the executive program.

In a redundant operating mode, the data presented to the VCS must be in sync to within a word time. To compensate for factors such as drift in the clocks, machine errors (e.g., parity), and other events, the IOP's perform a re-synchronizing operation after a specified number of I/O cycles. This re-synchronizing is accomplished primarily by hardware contained in the IOP. When re-synchronization is about to take place, the IOP's send a Master Sync Control Word to each other over the Type 1 I/O channels. Special hardware times the receipt of the control words, detects any computer clock failures by determining if one has drifted out of tolerance with respect to the majority, and determines the time of synchronization. The method of synchronization eliminates a hard core clocking system.

A description of the I/O process for outputting data to the external subsystems when operating in a three-way voting mode is shown in Figure 6. Computers 1, 2, and 3 are shown as operating in a voting mode, computer 4 could be performing another task or in a failed state. The VCS receives threee copies of the message and transmits the majority on the bus. Simultaneously, the IOP's of computers 1, 2, and 3 monitor the transmitted message by the "loop around" design of the bus. This monitoring process consists of comparing the information sent to the VCS with that transmitted on the buss, any discrepancies are noted, and a "go" "no-go" code is sent to the VCS by each IOP at the end of the monitoring process. Once again the majority "go/no-go" opinion would be transmitted on the bus. The IOP's also monitor the go/no-go code transmitted on the bus to determine if they are in agreement with the majority. The IOP's contain automatic message re-transmission capabilities in case of a majority no-go opinion. Based on the monitored go/no-go code, the IOP's either proceed to the next message or re-transmit the last message. This re-transmission is attempted twice with the same VCS, if three successive transmissions fail, the IOP's then switch over to a new VCS. Three successive transmission failures with the new VCS will result in the IOP's proceeding on to the next message in the IOP sequence. The IOP also stores status words at the end of each message which indicate the occurrence of any errors or retries in the transmission process.

Figure 7 indicates the process of inputting data from external subsystems for the same mode of operation, namely, three-way voting. The command to input data to the computer systems is sent to the external subsystem in the same manner as described above for data output, namely, over one bus. However, the data is sent to the computer system over multiple buses. Further, the data received over a bus is sent to all the IOP's; therefore, each

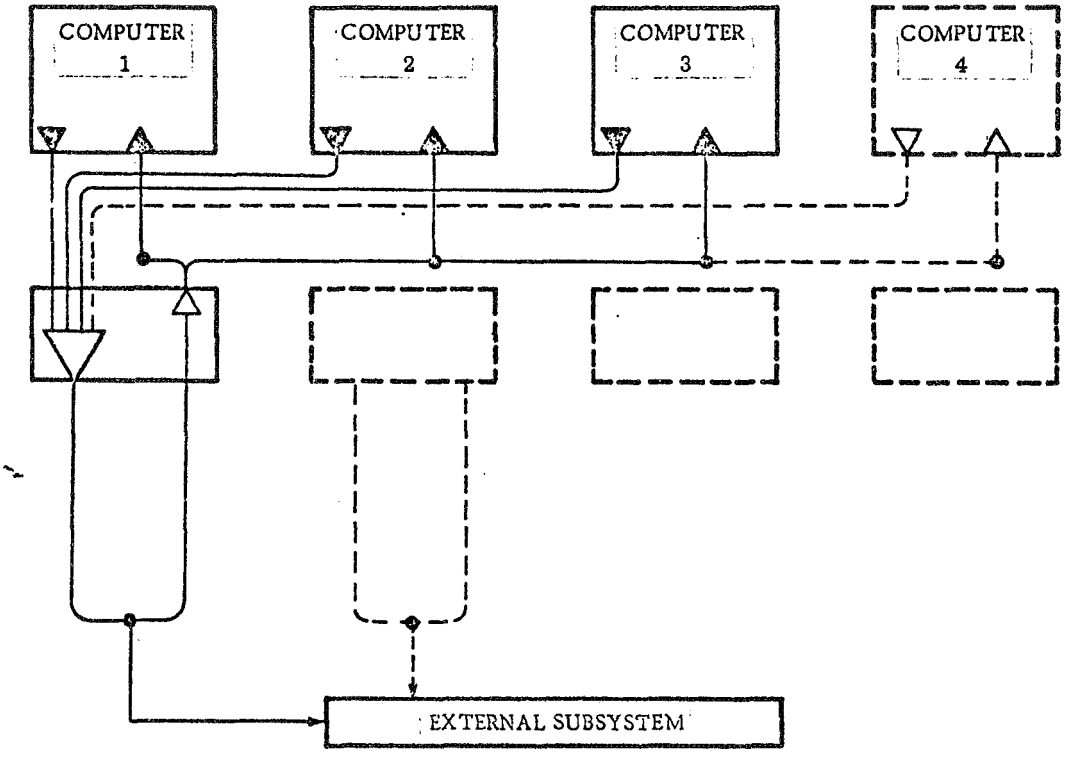


Figure 1-6. DATA OUTPUT PROCESS

DATA INPUT PROCESS

- 3 COPIES OF AN "INPUT REQUEST" ARE TRANSMITTED TO A SINGLE VCS.
- THE INPUT REQUEST IS TRANSMITTED IN THE SAME MANNER AS OUTPUT MESSAGES.
- THE INPUT DATA REPLY IS RECEIVED OVER MULTIPLE BUSSES.
- THE MULTIPLE COPIES ARE COMPARED USING A SOFTWARE VOTING PROCESS.

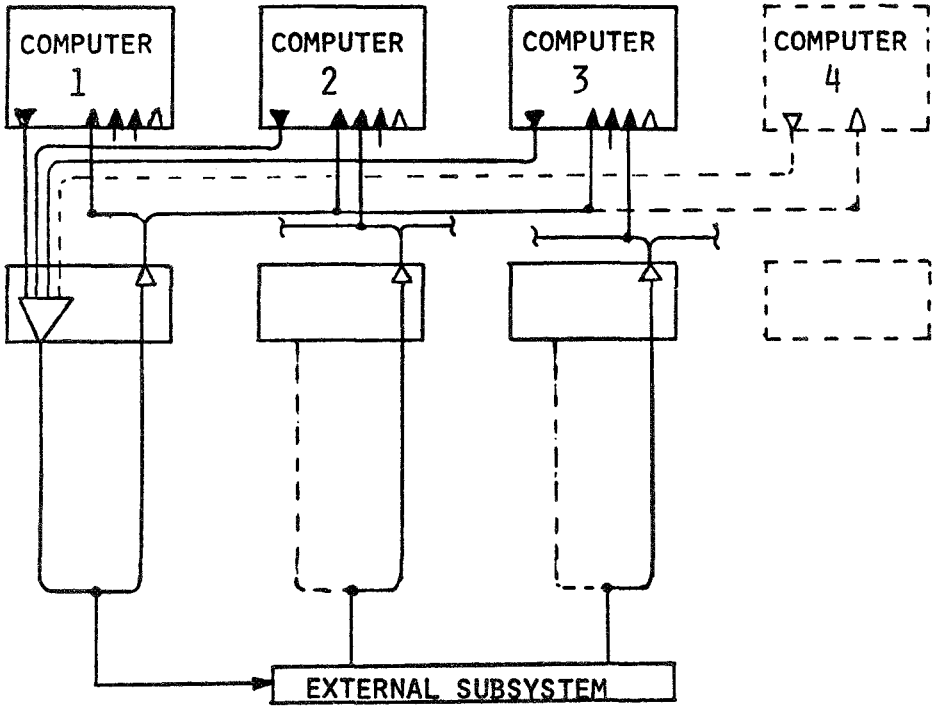


FIGURE 1-7. DATA INPUT PROCESS

- IOP will receive three copies of the data sent to the computer system. This redundant set of input data received by each IOP is used in a software voting process by the CPU. Preceding the transmission of data, the external subsystem transmits an acknowledge word. The IOP's all check the command sent to the external subsystem (by the loop-around feature of the bus) and the acknowledge word before accepting the input data. This check also serves to determine if a retransmission is to be attempted.

2. DETAILED DESCRIPTION OF IOP/VCS FUNCTIONS

2.1 VCS Operation and Description

In Section 1.2 the basic architecture and overall block diagram of the VCS (Figure 1-3) was presented. This section presents the design of this device by discussing the control unit and voter-comparator-selector logic as shown in Figure 2-1.

2.1.1 VCS Control Unit

a) P Matrix

The function of the VCS control unit is to connect the VCS output unit appropriately to receive data from the four computers. Another function of the control unit is to determine which computers are good or bad. The control unit consists of two basic functional elements: The P matrix and the R matrix. The P matrix contains information on the good and bad state of the four computers, while the R matrix contains the desired operating mode of the four computers (4V, 3V, 2CO selector). The P matrix is described below:

	A	B	C	D
A	AA	AB		
B				
C				
D				

It is seen to be a 4 x 4 matrix. The diagonal elements AA, BB, CC, DD are the prime information desired from the P matrix (this is what the R matrix uses). These elements define whether a computer is good or bad (if AA = 1, Computer A is good). The off diagonal elements AB, BA, etc., are one computer's opinion of another computer, i.e., AB is computer A's opinion of computer B. In general:

(i,j) = i's test of j

= 1 if i tests j to be good

= 0 if i tests j to be bad

The off diagonal elements (k,j) are directly input to the P matrix from the computers themselves while the diagonal elements are derived from logic associated with the P matrix.

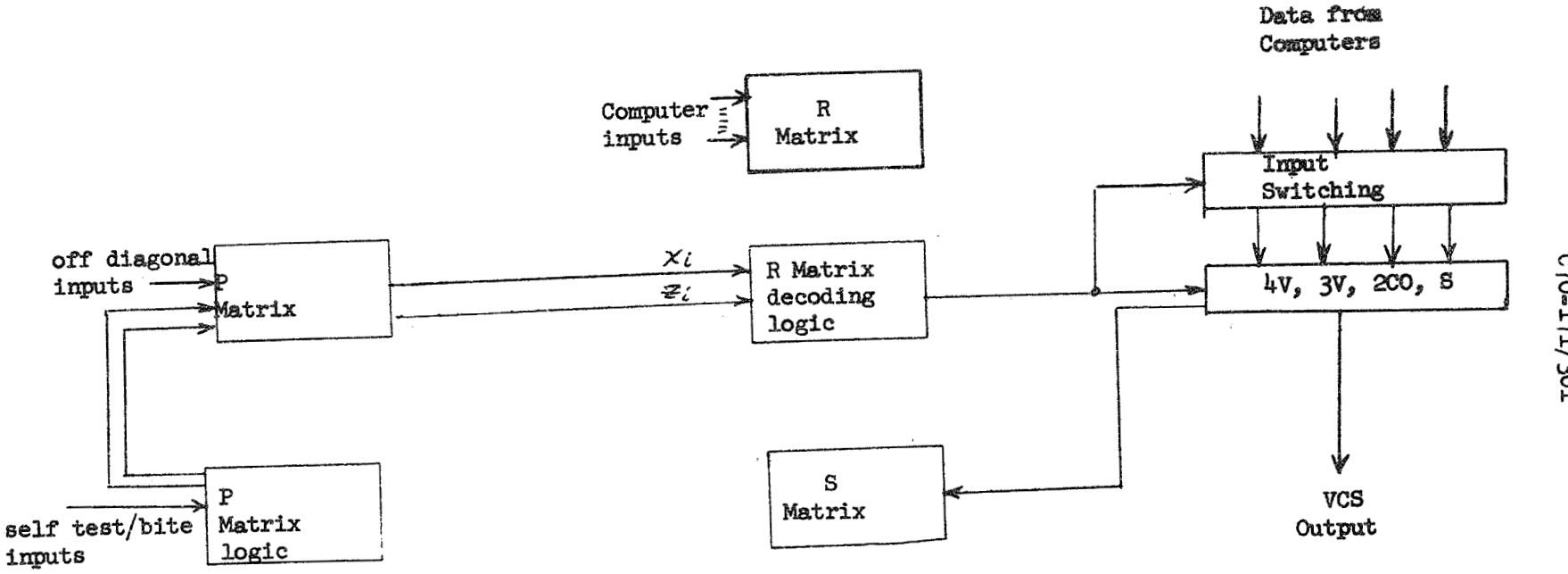


Figure 2-1. VCS Implementation

The logic that derives the diagonal elements will be explained below. The basic criteria for declaring a computer good or bad is as follows: A computer is good until either it reports itself as bad (self test/bite signal from computer = 0) or a majority of the other good computers think it is bad. The equations that are used to derive the diagonal element will be given below. Those for computer "D" will be given; the equations for computers A, B and C follow directly.

The register that contains the "D "column of the P matrix, AD, BD, CD, DD, is as shown in Figure 2-2.

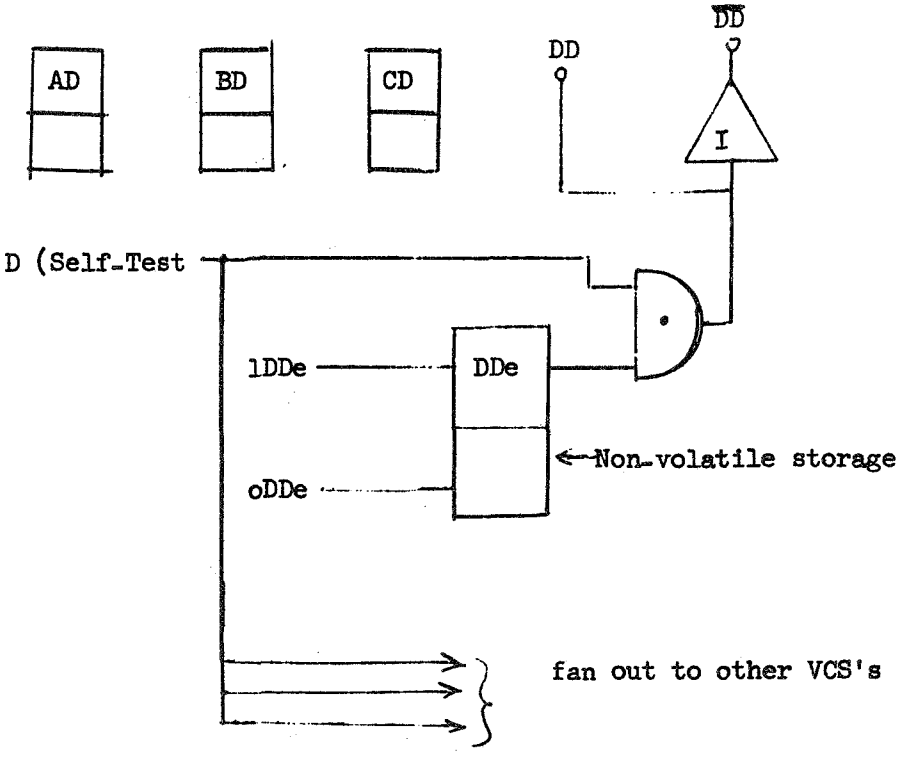


Figure 2-2. P Matrix

The flip flops that contain the terms AD, BD, CD are set directly by each of the computers A, B and C respectively. The term DD is derived as shown in the diagram. The term "D" represents the self test/BITE signal received from computer D. This signal is "anded" with the contents of DDe, the enable DD flip flop. The one set and zero set (1DDe and ODDe) terms for DDe are given below:

$$\begin{aligned} \text{ODDe} = &(AA)\ (BB)\ (\underline{AD})\ (\underline{BD}) \\ &+(AA)\ (CC)\ (\underline{AD})\ (\underline{CD}) \\ &+(BB)\ (CC)\ (BD)\ (CD) \end{aligned}$$

$$\begin{aligned} \text{1 DDe} = &(AD)(BD)(CD)(AA)(BB)(CC) \\ &+(BD)(CD)(\overline{AA})(BB)(CC) \\ &+(AD)(CD)(AA)(\overline{BB})(CC) \\ &+(AD)(BD)(AA)(BB)(\overline{CC}) \\ &+(AD)(AA)(\overline{BB})(\overline{CC}) \\ &+(BD)(\overline{AA})(BB)(\overline{CC}) \\ &+(CD)(\overline{AA})(\overline{BB})(CC) \\ &+(\overline{AA})(\overline{BB})(\overline{CC}) \end{aligned}$$

The ODDe term represents the adaptive logic of the P matrix. This allows the system to adapt or mask out failed computers from decision processes. Note that the DD flip flop must actually be non-volatile storage since its condition cannot be lost due to any transients. This results since the adaptive logic can only handle single failures at a time and an attempt to re-establish the terms AA, BB, CC, DD after more than one failure can not be guaranteed.

The 1DDe term shows the condition wherein the set of DDe is accomplished after repair of the system. Note that 1DDe cannot be derived from the inverse of ODDe, an attempt to do so could cause the system to "blow up" after three failures. (After three failures the ODDe will go false and, if 1DDe were derived from this, self test/bite would have to be relied upon from all the failed computers)

The output from the P matrix logic to the R matrix logic is the good/bad indication of the four computers. If we let

γ_i = good state of computer i

ζ_i = bad state of computer i

Then for computer D

XD = DD

ZD = $\overline{DD}$

The situation when one computer fails, reports the other computers as bad, but fails to report itself as bad should be mentioned here. Suppose this has happened to computer A, the resultant configuration of the P matrix is:

	A	B	C	D
A	1	0	0	0
B	BA	1	1	1
C	CA	1	1	1
D	DA	1	1	1

If this is the first failure in the system, then BA, CA and DA would be set to 1. The terms $A\beta$, AC, and AD being 0 would have no effect on the diagonal elements of the P matrix since the logic requires a majority opinion as explained above. Since A is the bad computer, it is up to computers B, C and D to insert 0's in BA, CA, and DA (only 2 out of the three are required); once this is accomplished, the term AA will be forced to a zero.

Note that after two failures, it may not be possible to reach a majority opinion in the logic associated with the P matrix. For the third failure, the primary term relied on is the self test/bite indication.

b) R matrix

The output unit of the VCS shown in Figure 2-3 has the capability of acting as a 4 input voter, 3 input voter, 2 input comparator, or a selector switch on the outputs of the 4 sets of triple shift registers.

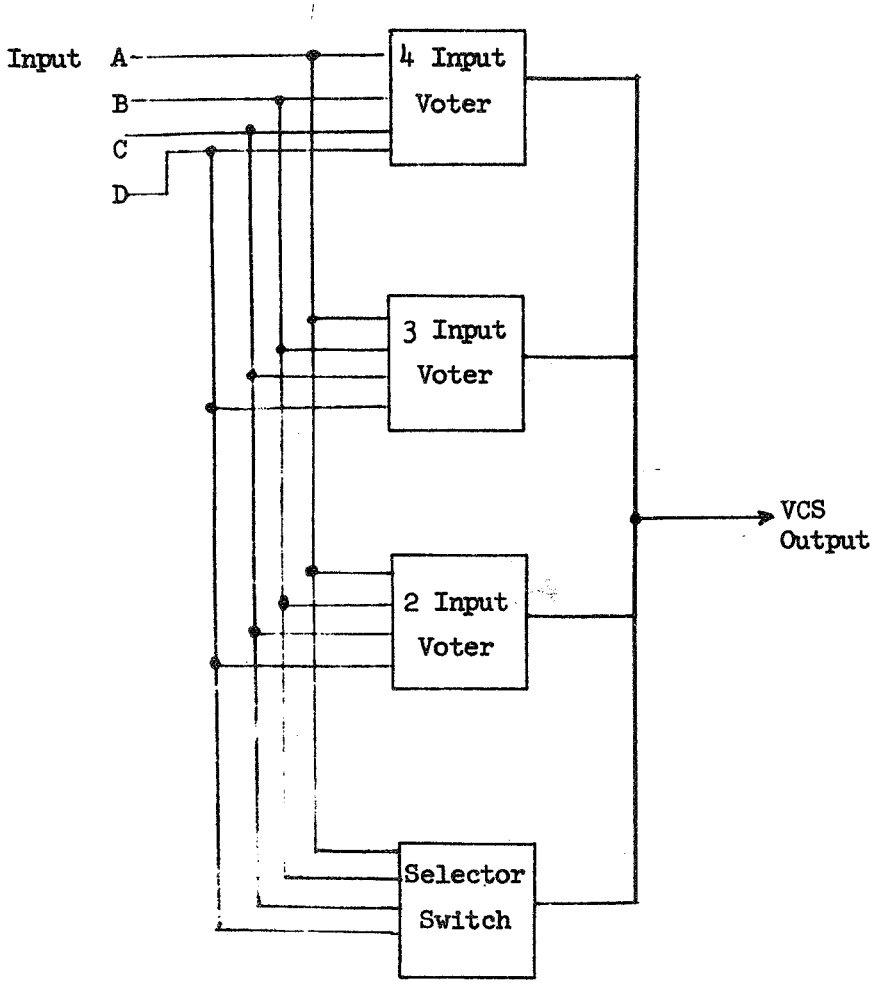


Figure 2-3. Voter-Comparator-Selector

The method of switching these inputs is by means of the R matrix. The R matrix switches the appropriate voter, comparator or selector to receive inputs from the selected computer(s). The R matrix coupled with the results of the P matrix (X_i, Z_i) then allows the VCS to function in a particular mode. For example, the R matrix would be set to:

	A	B	C	D
A	1	1	1	1
B	1	1	1	1
C	1	1	1	1
D	1	1	1	1

if the VCS is to work as a 4 input voter, as:

	A	B	C	D
A	1	1	1	
B	1	1	1	
C	1	1	1	
D				

if the VCS is to function as a 3 input voter between computers A, B, C.

No ambiguity should be presented by the R matrix, for example:

1	1	1	
1	1	1	
1	1	1	
			1

would represent a conflict to the particular VCS - that is, whether to operate as a 3 input voter on computers A, B, C or as a selector outputting computer D.

The R matrix decoding logic is designed such that the majority of the good computers (as defined by X_i, Z_i) must agree on a particular mode for that mode to be selected by the R matrix. A computer that will not be participating in that particular mode is required to insert all o's in its particular row. This essentially represents a don't care condition; i.e., it will go along with whatever mode the others want to operate in.

The R matrix is decoded as follows for a 4 input voter:

$$\begin{aligned} 4V/ABCD = {} & (rA_{15})(rB_{15})(rC_{15})\ X_A\ X_B\ X_C \\ & + (rA_{15})(rB_{15})(rD_{15})\ X_A\ X_B\ X_D \\ & + (rA_{15})(rC_{15})(rD_{15})\ X_A\ X_C\ X_D \\ & + (rB_{15})(rC_{15})(rD_{15})\ X_B\ X_C\ X_D \end{aligned}$$

Where the nomenclature here is:

r_{ij}; i, j = A, B, C, D representing an element in the R Matrix

r_{ik}; i = A, B, C, D k = numeric 0 → 15 representing decoded condition of ith row.

A three input voter between computers A, B, C would be decoded as follows:

$$\begin{aligned} 3V/ABC = {} & (rA_{14})(rB_{14})(rC_{14})\ X_A\ X_B\ X_C \\ & + (rA_{14})(rB_{14})\ X_A\ X_B\ (rD_o + Z_D) \end{aligned}$$

$$+ (rA_{14})(rC_{14})\ X_A\ X_C\ (rD_o + Z_D)$$

$$+ (rB_{14})(rC_{14})\ X_B\ X_C(rD_o + Z_D)$$

Similar conditions apply for 3V/ABD, 3V/ACD, 3V/BCD.

A two input comparator between A and B would be decoded as follows:

$$2CO/AB = (rA_{12})(rB_{12})\ X_A\ X_B\ (rC_o + Z_c)$$

$$+ (rA_{12})(rB_{12})\ X_A\ X_B\ (rD_o + Z_D)$$

Similar conditions apply to 2CO/AC, 2CO/AD, 2CO/BC, 2CO/BD, 2CO/CD

A single input selector from A would be decoded as follows:

$$S(A) = (rA_8)\ X_A\ (rB_o + Z_B)(rC_o + Z_c)(rD_o + Z_D)$$

Similar conditions apply to S(B), S(C), S(D)

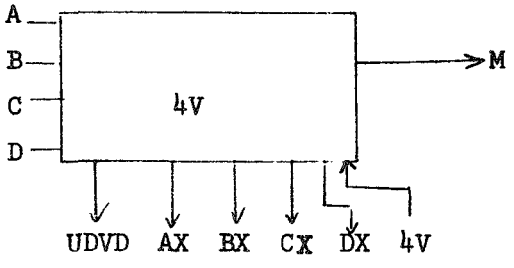
The additional proviso pertains to S(A) and that is X_A must be anded to the equation. The state of the R matrix must be decoded to obtain:

4V
3V (ABC)
3V (ABD)
3V (ACD)
3V (BCD)

2CO (AB)
2CO (AC)
2CO (AD)
2CO (BC)
2CO (BD)
2CO (CD)

S (A)
S (B)
S (C)
S (D)

These signals then enable the particular voter/comparator/switch and determine which lines (computers) are connected. The four input voter block diagram is:



Where

M is the voted output

AX, BX, CX, DX indicate a discrepancy in lines A, B, C, D respectively, UDVD indicates an undecidable voter discrepancy

$$M = ABC + ACD + ABD + BCD$$

$$AX = \bar{A}BCD + A\bar{B}\bar{C}\bar{D}$$

$$BX = A\bar{B}CD + \bar{A}B\bar{C}\bar{D}$$

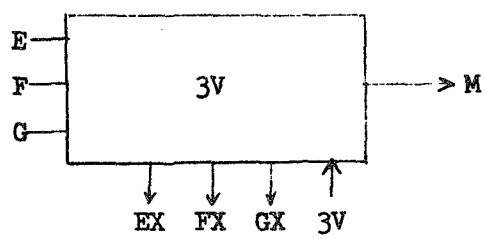
$$CX = AB\bar{C}D + \bar{A}\bar{B}C\bar{D}$$

$$DX = ABC\bar{D} + \bar{A}\bar{B}\bar{C}D$$

$$UDVD = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + ABC\bar{D}$$

where the terms A, B, C, D in the above equations are necessarily anded with 4V.

The three input voter becomes:



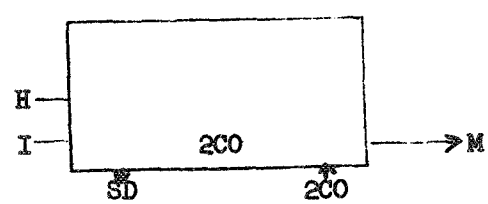
$$M = EF + EG + FG$$

$$EX = E\bar{F}\bar{G} + \bar{E}FG$$

$$FX = E\bar{F}G + \bar{E}F\bar{G}$$

$$GX = \bar{E}\bar{F}G + EF\bar{G}$$

And the two input comparator:



Where $M = H\bar{I}$

$SD = \bar{H}I + \bar{H}\bar{I}$

The switching of the input lines ABCD to the lines E, F, G, H, I is accomplished by means of the switching network:

$$E = A.3V(ABC) + A.3V(ABD) + A.3V(ACD) + B.3V(BCD)$$

$$F = B.3V(ABC) + B.3V(ABD) + C.3V(ACD) + C.3V(BCD)$$

$$G = C.3V(ABC) + D.3V(ABD) + D.3V(ACD) + D.3V(BCD)$$

$$H = A.2CO(AB) + A.2CO(AC) + A.2CO(AD) + B.2CO(BC) + B.2CO)BD) + C.2CO(CD)$$

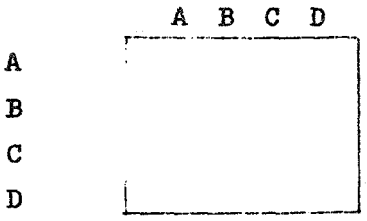
$$I = B.2CO(AB) + C.2CO(AC) + D.2CO(AD) + C.2CO(BC) + D.2CO(BD) + D.2CO(CD)$$

and the selector is switched by

$$M = A.S(A) + B\ S(B) + C\ S(C) + D\ S(D)$$

c) S Matrix

The S matrix contains the error status of input data to the voting and comparison logic discussed above (4V, 3V, 2CO):



The S matrix is a 4 X 4 matrix, a 1 X 4 matrix is all that is required to indicate any errors in data from computers A, B, C or D. However, to allow the computers to reset the matrix to zero, the row (1 X 4) is repeated three times thereby resulting in a row for each computer. Therefore

$$1,j = 2,j = 3,j = 4,j$$

$$1,j = \text{any errors in data from computer } j$$

For computer A, j = 1 and from the above discussion.

$$1,\ 1 = AX + UDVD + EX\ [3V(ABC) + 3V(ABD) + 3V(ACD)] + SD[2CO(AB) + 2CO(AC) + 2CO(AD)]$$

It can be seen that in a voting mode the logic will detect which computer disagreed with the majority, whereas in a comparison made the logic will set two bits if a discrepancy exists.

2.1.2 VCS Communication

The VCS contains three matrices as described above. Each of these matrices may be sampled under command from the computers. All 16 bits of the matrix are sent to the computer if sampled. The computers can also set the matrices. However, the elements that they can set are limited:

P matrix: each computer can set the non-diagonal elements in its row

R matrix: each computer can set its row

S matrix: each computer can reset its row

2.1.3 VCS Mechanization

The above discussion presented the basic philosophy of the VCS device. Figure 2-4 contains a general block diagram indicating the interaction of the various elements discussed. The triple buffer registers shown in Figure 2-4 will be explained further. Figure 2-5 contains the detailed operation of this section. The "In Buffer" is part of the IOP and is the receive channel from one of the other computers (three are required). The received information is shifted into the input channel buffer and is decoded by logic in the IOP. Data and information destined for the VCS is sent in parallel to the set of triple buffer registers R_1, R_2, and R_3. It is sent to R_1 if a voter or comparator mode has been selected. When the next word is ready to be loaded into R_1 the present contents of R_1 are transferred to R_2 and likewise the present contents of R_2 would be transferred to R_3. Thus R_1, R_2, and R_3, act as three word times of buffering.

R_2 and R_3 also are shift registers that can output a serial stream of information to the VCS output logic. The decision as to when to begin shifting out to the VCS is made by logic that performs the function listed in the lower part of Figure 2-5. The control as to which data to output is derived from the R matrix. The decision logic as to when to shift out to the VCS allows for up to two word times out of sync. Normal operation (all computers good) requires that the data be in sync within one word time. The two word time tolerance thereby does not allow a bad computer from affecting the normal operation (e.g., a bad computer could be one word time ahead of a good computer and a second good computer could be a word time behind the other good computer). If a computer is greater than one word time out of sync with any other, then it is automatically decided as bad by control logic in the VCS. The detailed timing and control sequence for the VCS is presented in Section 4.

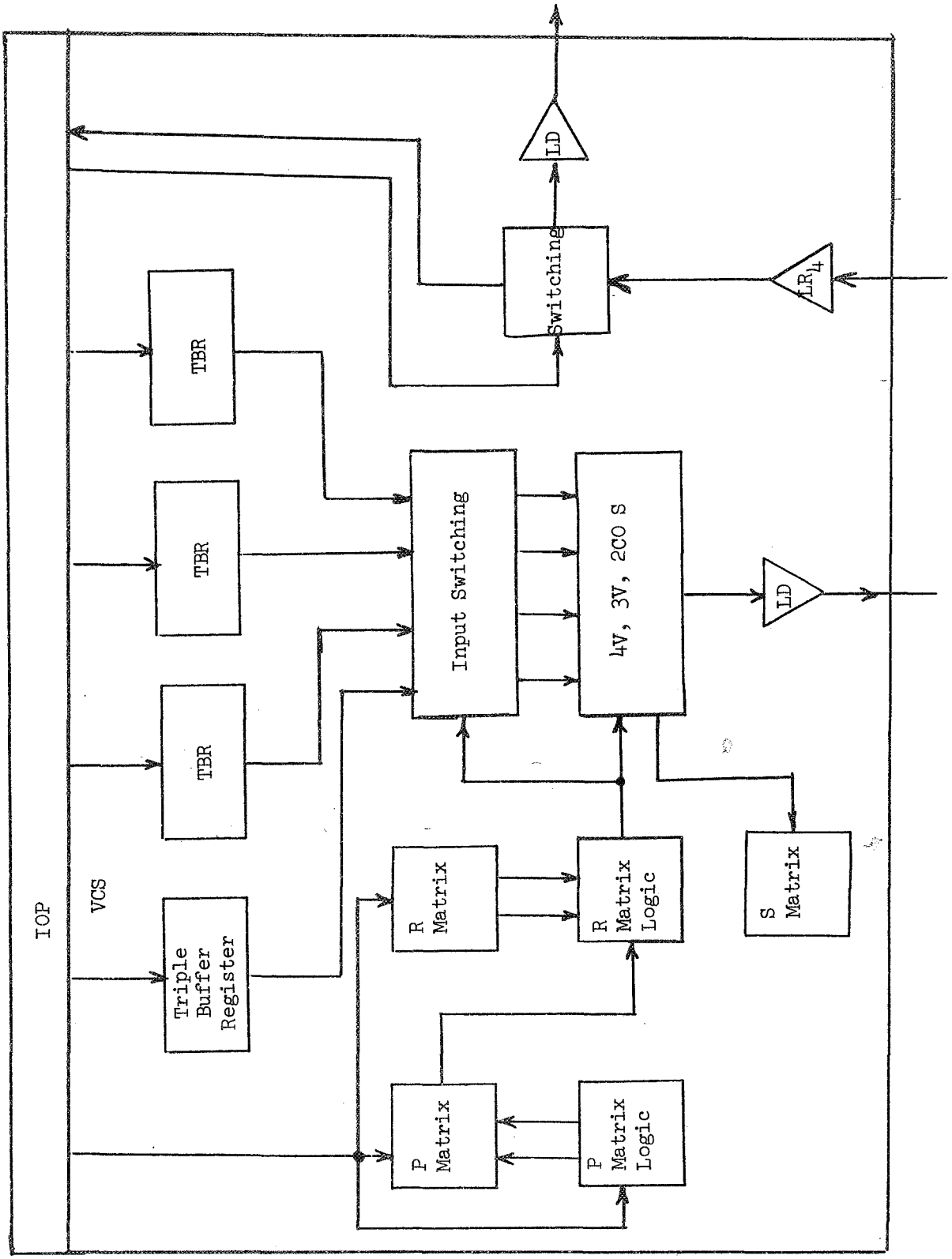


Figure 2-4. VCS Mechanization

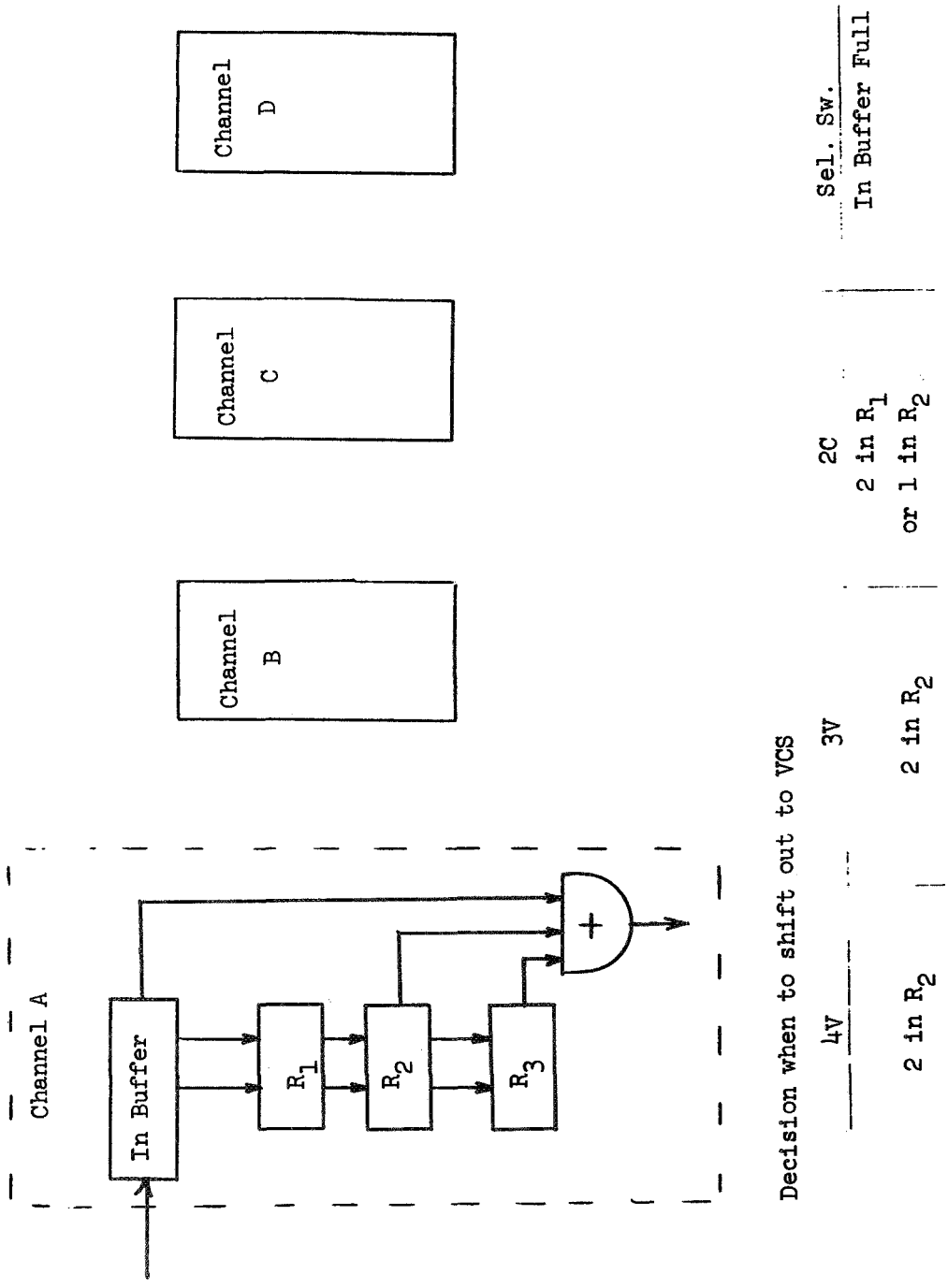


Figure 2-5. Triple Buffer Register

2.2 .0 IOP COMMANDS AND CONTROL WORDS

2.2 .1 Introduction

The IOP is set into operation by the decoding of a command and/or a control word. The commands are stored in the computer memory and are accessed according to the command program counter in the IOP. Control words are also stored in the memory and are also received from other computers in the system over the inter-computer bus. Commands are executed in sequence as any other software program. Control words are executed only when specified by a command or when a control word is received over the inter-computer bus.

The command and control word formats are given in Figures 2-6 and 2-7 respectively. Reference to these figures will clarify the following discussion.

2.2.1.1 Commands

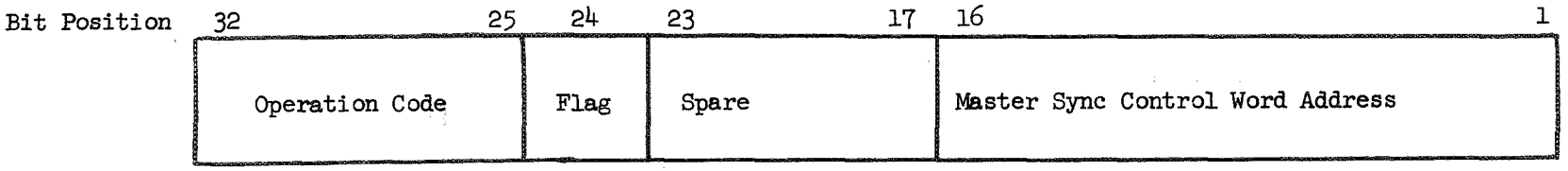
The IOP command list is as follows:

1. Halt and Proceed
2. Jump
3. Conditional Skip
4. Fetch Control Word

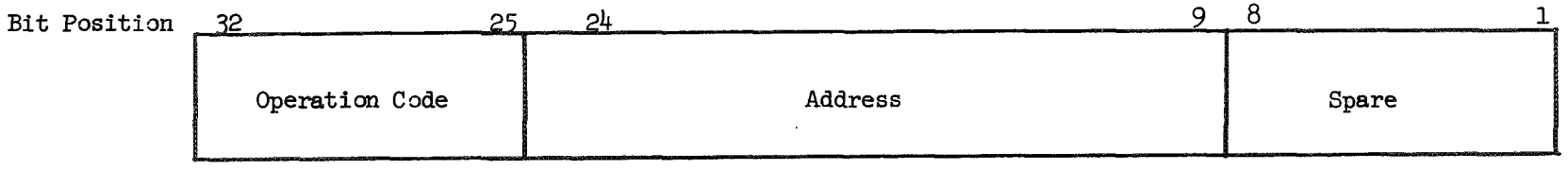
2.2.1.1.1 Halt and Proceed Command

The halt and proceed command (HPR) causes the IOP to enter an idle mode where no commands are executed. The IOP will respond to control words received over the inter-computer bus. If the flag bit is zero set, the IOP does nothing during the idle mode. If the flag bit is one set, the IOP performs a master sync operation when the real time counter reaches a specified count. In either case, when the idle mode is entered, the command program counter is set to a fixed code for accessing the next command when the idle mode is terminated.

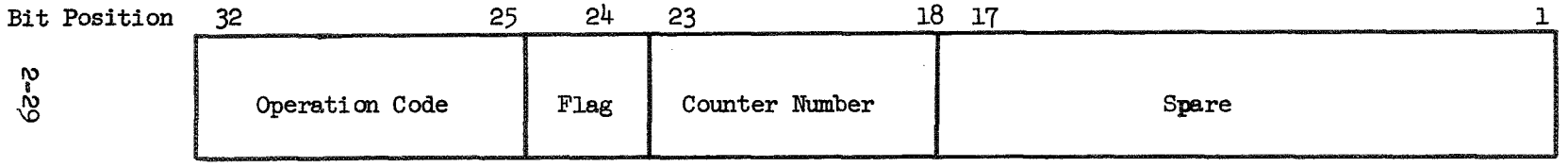
The master sync operation consists of sending a master sync control word to all other computers in the system and a master sync mask code



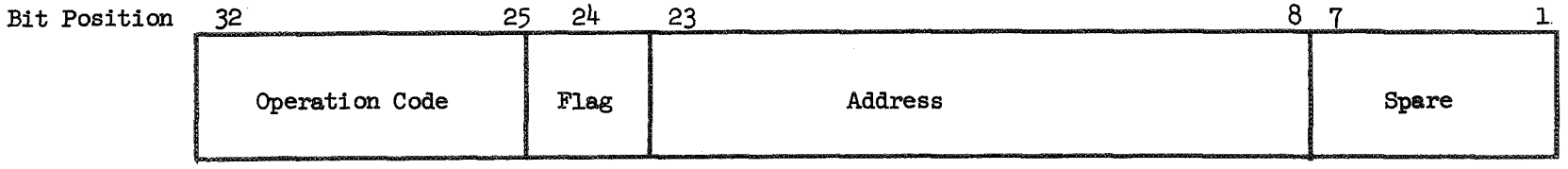
Halt and Proceed



Jump

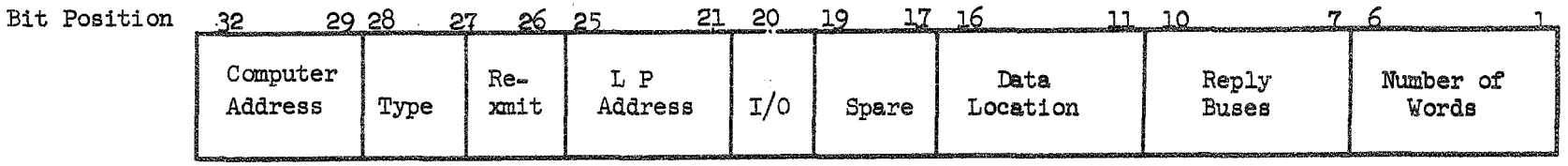


Conditional Skip

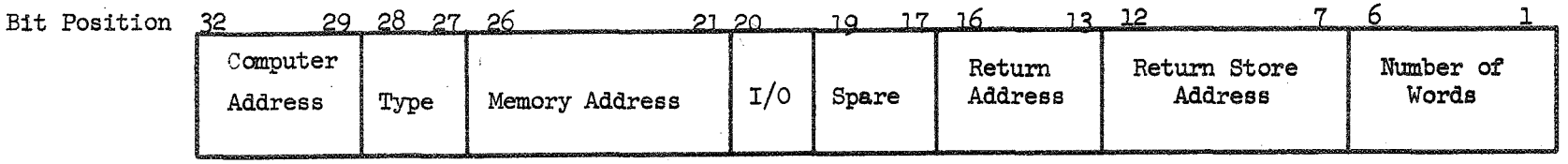


Fetch Control Word

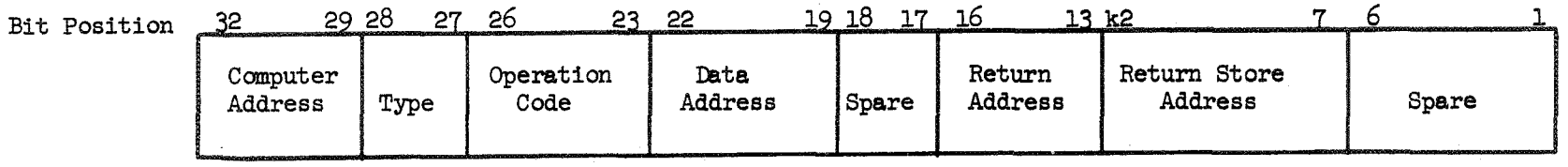
Figure 2-6. IOP Command Formats



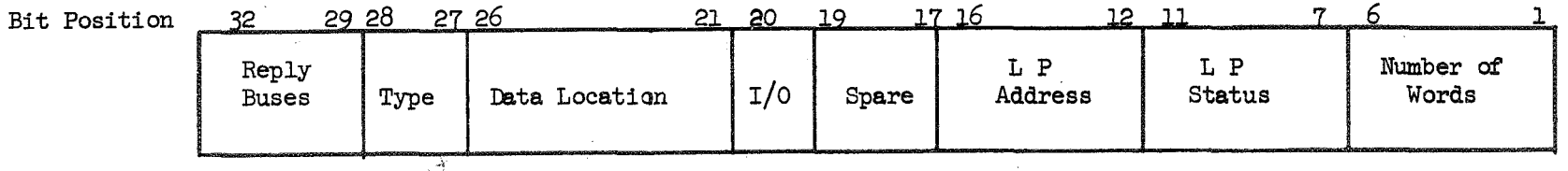
Computer to Local Processor Control Word



Computer to Computer (Memory) Control Word



Computer to Computer (IOP) Control Word



Acknowledge Control Word

Figure 2-7. CONTROL WORD FORMATS

to the IOP master sync controller. The master sync mask code is a four bit field in the master sync control word. The master sync control word is located in memory at the address specified by the sixteen least significant bits of the HPR command. The rest of the master sync operation consists of operating on the master sync control words as they are received by the IOP. A detailed description of this operation is given in the section on the master sync controller.

The IOP stays in the idle mode until either the real time counter interrupt or the master sync controller cycle start discrete goes true. The real time counter interrupt is used if the flag in the HPR command is zero set and the cycle start discrete is used if the flag bit is one set. The IOP accesses the memory according to the command program counter.

2.2.1.1.2 Jump Command

The jump command causes the IOP to use the address field of this command as the location of the next command to be accessed. The address field is placed in the command program counter and the memory accessed.

2.2.1.1.3 Conditional Skip Command

The conditional skip command causes the IOP to branch in the program depending upon certain conditions. Step by step operation of the command is as follows:

1. The counter number field is used as the least significant bits of a memory address. The address is placed in the operand program counter.
2. The contents of that location are read into a buffer register in the IOP.
3. The 16 least significant bits of the buffer register are treated as a counter and incremented by one count - any overflow is noted and stored.

4. If an overflow is detected:

   (a) The 16 most significant bits of the buffer register are copied into the 16 least significant bit positions. This re-initializes the counter.

   (b) The buffer register contents are stored back in the same memory location they came from.

   (c) The command program counter of the IOP is incremented one count.

5. If no overflow is detected, the flag bit of the command is investigated.

6. If the flag bit is a zero:

   (a) The contents of the buffer register are placed back in the same memory location they came from.

   (b) The command program counter is incremented by two counts.

7. If the flag bit is a one, the command program counter is incremented by two counts.

2.2.1.1.4 Fetch Control Word Command

The fetch control word command causes the IOP to place the address field of the command in the operand program counter. This location is accessed and the contents read into a storage register. This word is a control word and further operation of the IOP is controlled by this control word.

If the flag bit is zero set, all input/output buses of the IOP are allowed access to the computer memory. If the flag bit is one set, the input/output bus between the computer and the mass memory/data management system is locked out. This is done to preserve program synchronization during critical operating modes.

After the fetch control word command is successfully executed, the command program counter is incremented by one count and the next command is accessed.

2.2.1.2 Control Words

All data going in or out of the computer is controlled by the IOP executing control words. To output data or data requests over the Types 1 and 2 channels and to input data over the Type 2 channel, the IOP operates according to control words accessed from the computer memory. To input data or data requests over the Type 1 channel, the IOP operates according to control words received from an external source. There are four control words:

1. Computer to local processor control word
2. Computer to computer (memory) control word
3. Computer to computer (IOP) control word
4. Acknowledge control word

The IOP is capable of receiving all control word types from external devices, but can only originate the first three control words listed above.

The type of control word is specified by the type field (bits 27 and 28) of the control word as follows:

Type	Bits	
	28	27
Acknowledge	0	0
Computer to local processor	0	1
Computer to computer (memory)	1	1
Computer to computer (IOP)	1	0

2.2.1.2.1 Computer to Local Processor Control Word

This control word is used when data are to be sent to or requested from a local processor. The fields of the control word are defined as follows:

Field	Bits	Definition
*Computer Address	4	Indicates which computers are to receive the control word. Any combination of computers may be addressed by setting the appropriate bits to ONE.
Type	2	Itentifies control word type.
Retransmit	1	ONE: Message is to be sent again if an error is detected in the transmission. ZERO: Message is to be sent once.
LP Address	5	Identifies the local processor being accessed by the computer.
I/O	1	ONE: Data are to be input to the computer. ZERO: Data are to be output from the computer.
Data Location	6	Identifies the address of the first data word of the message in the local processor memory.
*Reply Buses	4	Identifies the buses over which data are to be transmitted from the local processor to the computer.
Number of Words	6	Indicates the number of data words in the message.

When the control word is first accessed from memory, the fields marked by an asterisk in the above list are not the codes having the meaning given in the definition column. The initial value of these fields is used as an address to get the actual codes to be used in the control word. The IOP places the initial values in the least significant bit positions of the operand program counter, sets the upper bits to a fixed code and accesses the memory. The word accessed from memory contains the

true codes which are inserted into the register holding the control word. The control word is ready to be transmitted.

2.2.1.2.2 Computer to Computer (Memory) Control Word

This control word is used when data are to be sent between computer memories over the inter-computer bus. The fields of the control word are defined as follows:

Field	Bits	Definition
Computer Address	4	Same as Para. 2.2.1.2.1
Type	2	Same as Para. 2.2.1.2.1
Memory Address	6	Identifies a location in the memory of the computer receiving the control word.
I/O	1	Same as Para. 2.2.1.2.1
Return Address	4	Identifies the computer sending the control word.
Return Store Address	6	Identifies a location in the memory of the computer sending the control word.
Number of Words	6	Same as Para. 2.2.1.2.1

2.2.1.2.3 Computer to Computer (IOP) Control Word

This control word is used when the R, P and S matrices of the other computers are to be sampled or set and when a master sync operation is to be performed. The fields of the control word are defined as follows:

Field	Bits	Definition
Computer Address	4	Same as Para. 2.2.1.2.1
Type	2	Same as Para. 2.2.1.2.1
Operation Code	4	Identifies the operation to be performed by the receiving IOP.
Data Address	4	Identifies the location in the memory of the transmitting computer of the data to be used in setting the matrices.

Field	Bits	Definition
Return Address	4	Same as Para.2.2.1.2.2
Return Store Address	6	Same as Para. 2.2.1.2.2

The data address field of the control word, when read from memory, forms the least significant bits of the location of the actual address to be used for accessing the data to be sent.

The operation field codes are as follows:

Bits: 26	25	24	23	Operation
0	0	0	0	Set R and P matrices to value; Set S matrix to zero.
0	0	0	1	Set R matrix to value.
0	0	1	0	Set P matrix to value.
0	0	1	1	Set S matrix to zero.
0	1	0	0	Sample R, P and S matrices.
0	1	0	1	Sample R matrix.
0	1	1	0	Sample P matrix.
0	1	1	1	Sample S matrix.
1	0	0	0	Master Sync.

2.2.1.2.4 Acknowledge Control Word

The control word is generated by the local processors and sent to the computer. The control word is sent to acknowledge the receipt of data by the local processor or to start the transmission of data by the local processor. The fields of the control word are defined as follows:

Field	Bits	Definition
Reply Buses	4	Same as Para. 2.2.1.2.1
Type	2	Same as Para. 2.2.1.2.1
Data Location	6	Same as Para. 2.2.1.2.1
I/O	1	Same as Para. 2.2.1.2.1
LP Address	5	Same as Para. 2.2.1.2.1
LP Status	5	Identifies the operational status of the local processor and errors in data received by the local processo
Number of Words	6	Same as Para. 2.2.1.2.1.

2.3 IOP Bus Operations

The operations over the type 1 and 2 data buses will be described in this section. Any one operation may involve one or both of these buses with the input and output channels of each used in various combinations. The possible operating modes are:

a. Output data to local processor with voting using internal VCS.

b. Output data to local processor with voting using external VCS.

c. Output data to local processor without voting.

d. Input data from local processor with voting using internal VCS.

e. Input data from local processor with voting using external VCS.

f. Input data from local processor without voting.

g. Request data from another computer.

h. Input data from another computer.

i. Output requested data to another computer.

j. Output data to another computer.

k. Set matrices in another computer.

l. Set own matrices according to data from another computer.

m. Sample matrices in another computer.

n. Sample own matrices on request of another computer and transmit the data to the other computer.

o. Master sync operation.

Reference to the block diagrams of Paragraph 3.0 and the discussion of command and control words in Paragraph 2.2 will aid in the understanding of the following descriptions.

2.3.1 Output Data to Local Processor

The three ways of outputting data to a local processor vary only in whether voting is done and where the voting is done. The settings of the P and R matrices determine which way the data are handled.

In all cases, the start of an output data operation is the execution of the fetch control word command getting the control word from memory and into the control word store register in the IOP command control section. The next sequential location after the one that holds the control word holds the memory address of the first data word of the message. This data word address is read from memory into the buffer register 2 of the IOP command control for later use.

Bits 20, 27 and 28 of the control word register are investigated next and, if set to a binary code of 010, this indicates an output data to local processor operation. Bits 29 through 32 of the control word register are transferred into the least significant bit positions of the operand program counter to form a memory address. This memory location is read into buffer register 1. This word contains three (3) four bit fields:

a. Primary computer address
b. Secondary computer address
c. Reply buses code

Depending upon the setting of the VCS switched indicator of the IOP command control, either the primary or secondary computer address fields are read from buffer register 1 into bit positions 29 through 32 of the control word register. The reply buses code field is read into bit positions 7 through 10 of the control word register.

The control word is now complete and ready to be transmitted. The contents of buffer register 2 are read into the operand program counter in preparation for getting the data words from memory. The number of words field (bits 1 through 6) of the control word register are read into the word counters associated with the type 1 output channel and the type 2 input and output channels. The control word is read from

the control word register into buffer register 2 in preparation for sending.

2.3.1.1 Voting Using Internal VCS

Data flow through the IOP during this operation is diagrammed in Figure 2-8. This operation involves the Type 2 input and output channels, the VCS, and Type 1 output channel, one to three Type 1 input channels and the IOP command control.

The control word in buffer register 2 is transferred in bit serial fashion into a VCS buffer register. The data from the other computers involved in the voting arrives over a Type 1 input channel and transferred into other VCS buffer registers. These registers are used to remove small time differences in the arrival times of the data. All data are fed to the voter whose output is read into VCS buffer register 4. The contents of this register are valid only if a majority of inputs to the voter have agreed for the full word on a bit by bit comparison basis. This majority approved data are then transmitted to the local processor over the Type 2 output bus.

As the data are shifted to the VCS buffer registers, the data are also placed in a comparison register. The data are held here temporarily for a check against the data that are being received at the local processor. As the local processor receives the data, the same data returns to the computer on the Type 2 input channels. The IOP receives the data, ships the data out on the Type 1 output channel and compares the data with that in the comparison register. Discrepancies in the comparison are stored in the comparison indicator of the Type 2 input channel. This comparison continues until the last data word has been compared.

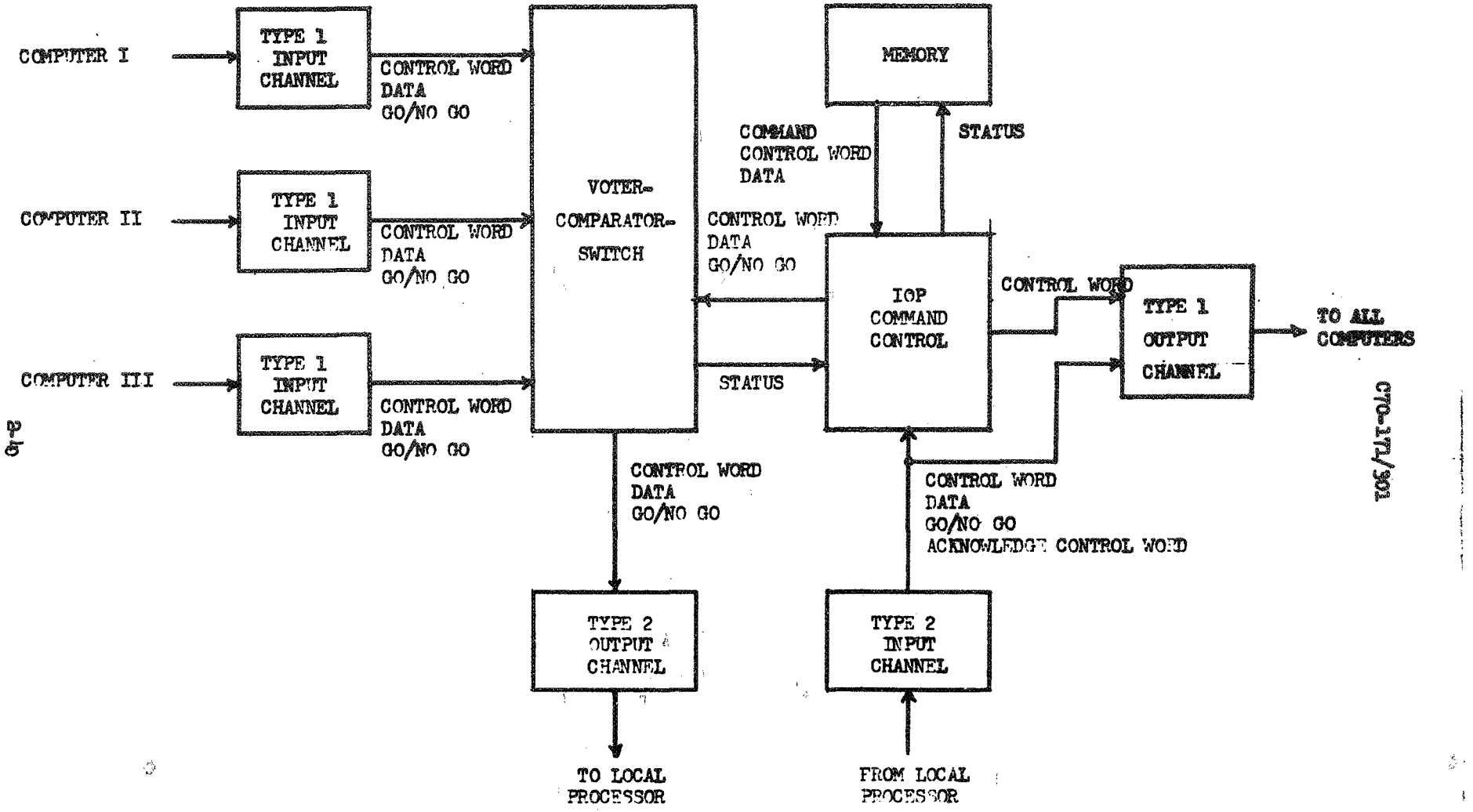


Figure 2-8.- OUTPUT DATA TO LOCAL PROCESSOR USING INTERNAL VCS

The local processor can use the received data only if a Go code is sent as the last word of the message. The sending of this Go code is determined by the involved computers and the results of their comparison of the data received back from the local processor and the data the IOP sent to the VCS. Each computer in the operations generates a Go or No-Go code based upon their comparisons and sends that code to the VCS. The code in the majority is sent to the local processor.

The code sent to the local processor is fed back to each computer. Indicators are set to indicate whether the IOP differed with the majority or not and which code was received by the local processor. The IOP waits for the arrival of the acknowledge control word from the local processor. A status word is formed in the IOP based upon the acknowledge control word, the Go or No-Go code sent and received and all other indicators generated during the message transmission. The status word is stored in memory in the location following the last data word.

With the status word stored away, the IOP checks the re-transmit flag (bit 26) of the control word register. If the flag is a binary zero, the operation is considered completed and the IOP command control goes on to the next operation. If the flag is a binary one and a No-Go code was received over the Type 2 input channel, the IOP command control increments the re-transmit counter. The next step depends upon the count of the re-transmit counter and the VCS switched indicator.

If the count is less than three, the IOP command control ignores the VCS switched indicator and goes to execute the command held in the command word store register. This is the same command that started the operation so the operation is repeated.

If the count is three and the VCS switched indicator is zero set, this means that three attempts have been made using the same VCS to send the data to the local processor. The VCS switched indicator is one set to indicate the next attempts are to be made using the secondary VCS assigned. The IOP command control then returns to repeat the operation.

If the count is three and the VCS switched indicator is one set, this means that three attempts were made to get the data to the local processor using the primary VCS assigned and three attempts using the secondary VCS assigned with no successful attempts. This is indicated by the status word stored away at the end of the last try. The IOP command control goes on to the next operation considering the other completed.

Voting involving three computers always gives a majority result if the VCS is operating correctly. However, in voting with two or four computers, there are cases where majorities may not exist. This is detected by the voter and causes the transmitter for the type 2 output channel to be turned off. The local processor responds to a loss of signal by resetting to the idle mode terminating the communication. This termination results in the loss of signal on the type 2 input channel and a consequent loss of signal on the type 1 output channel. All computers respond to this loss of signal by terminating the output operation and generating a status word as before. Retransmission is tried as described earlier.

2.3.1.2 Voting Using External VCS

Data flow through the IOP during this operation is diagrammed in Figure 2-9. This operation involves the use of the type 1 input channel associated with the computer doing the voting, the type 1 output channel and the IOP command control.

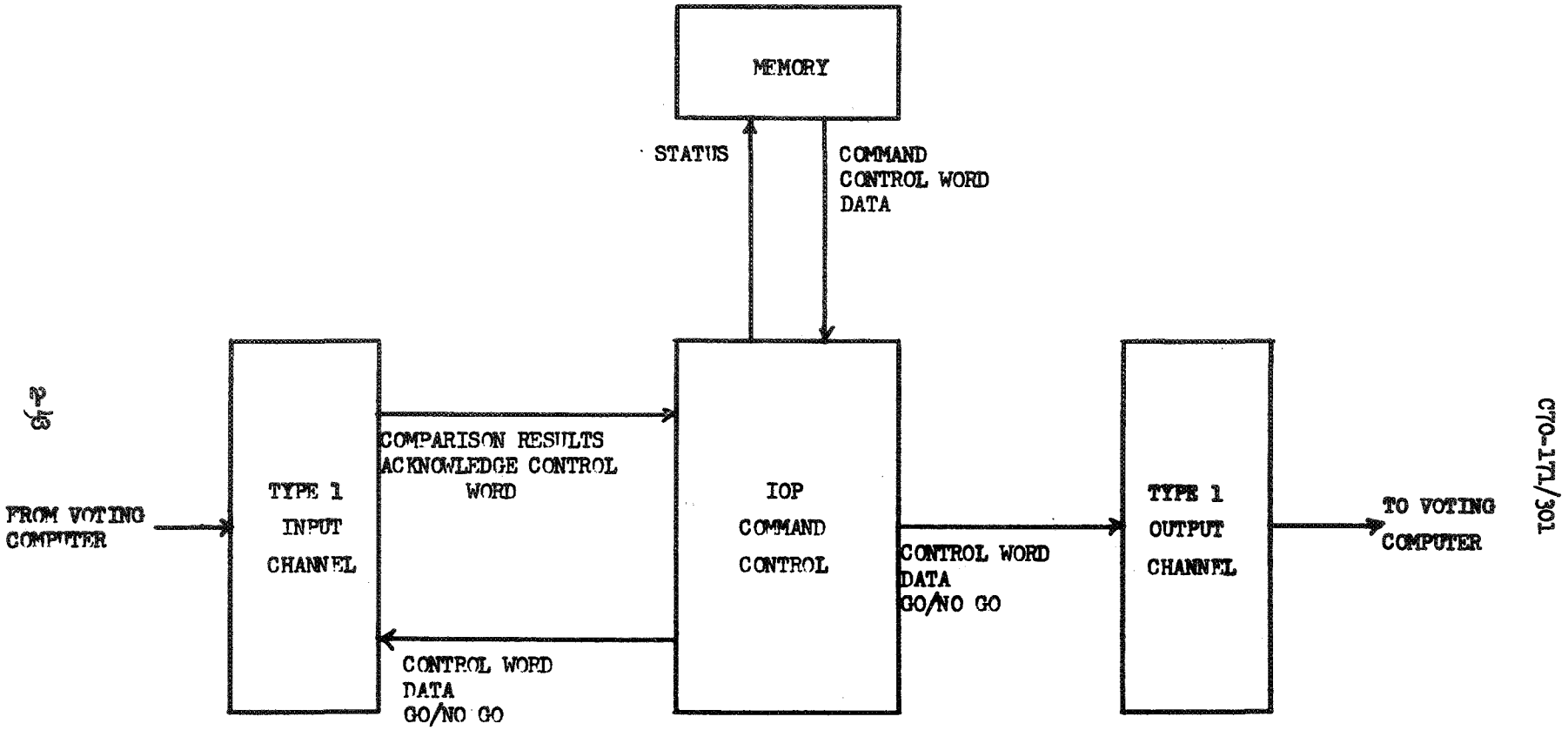


Figure 2-9. OUTPUT DATA TO LOCAL PROCESSOR USING EXTERNAL VCS

The control word held in buffer register 2 is sent serially into the type 1 input channel comparison register and is sent out over the type 1 output channel. As the control word is being shifted out of buffer register 2, the IOP command control accesses the memory and reads the first data words into buffer register 1. When the last bit of the control word is shifted out of buffer register 2, the contents of buffer register 1 are read into buffer register 2. The shifting of the contents of buffer register 2 continues as for the control word. The word counter is decremented for each 16 bits of data sent out. Until the word counter reaches zero, the IOP accesses data words from memory.

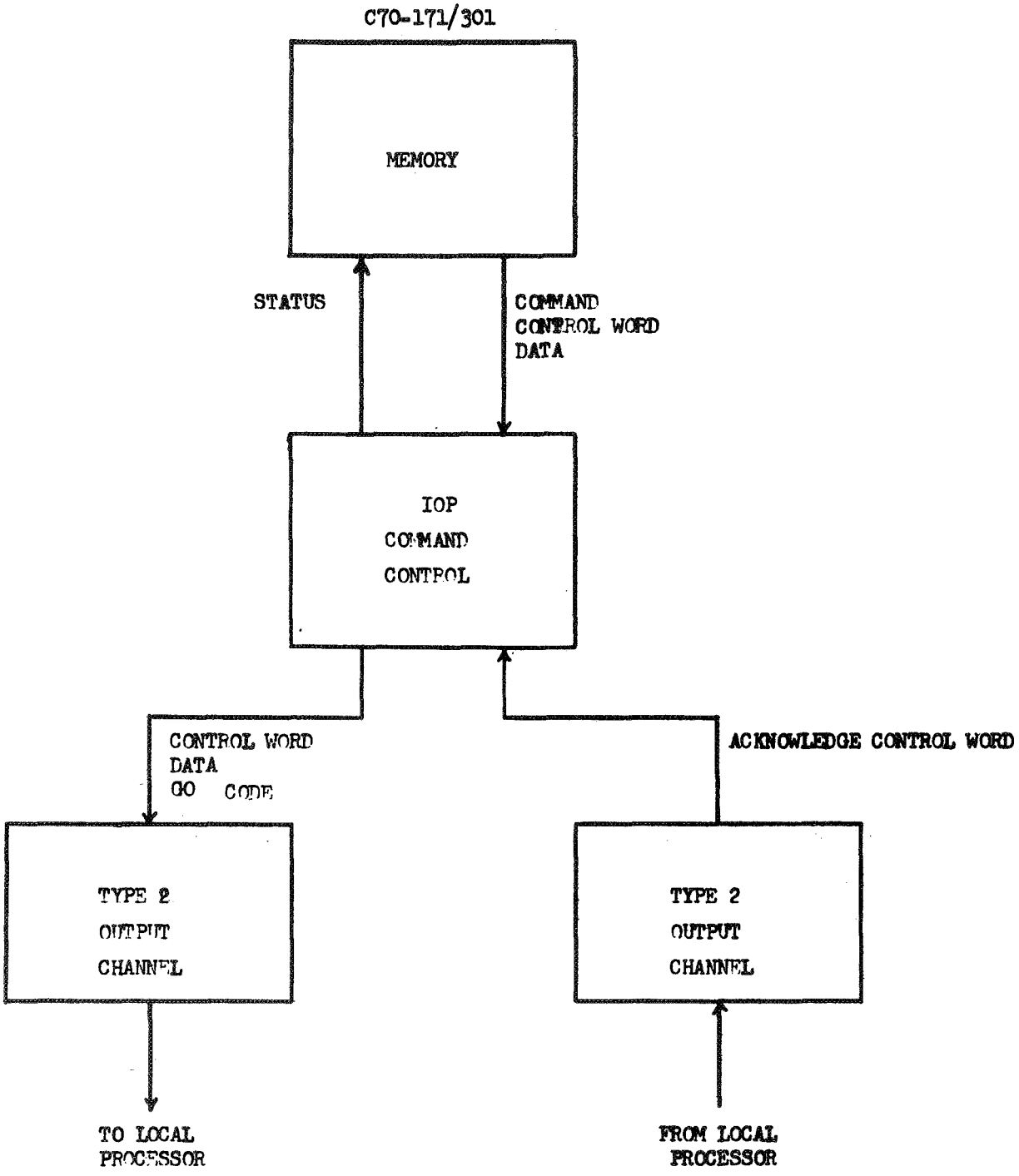
The voting computer sends the data returned via the Type 2 input to the IOP over the type 1 input channel. There it is compared with the data in the comparison register and a Go or No-Go generated and sent out. The results of the comparison are stored temporarily for inclusion in the status word.

After receiving the acknowledge control word, the IOP generates the status word and decides upon retransmission as described in Paragraph 2.3.1.1. After completion of the operation, either successfully or unsuccessfully, the IOP command control goes on to the next operation.

2.3.1.3 No Voting

Data flow through the IOP during this operation is diagrammed in Figure 2-10. The operation makes use of the type 2 input and output channels and the IOP command control.

In this operation, the IOP communicates directly with the local processor with no other computer involved. The control word in buffer register 2 is sent out over the type 2 output channel. As the control word is transmitted, the first data words are accessed from



OUTPUT DATA TO LOCAL PROCESSOR WITHOUT VOTING

FIGURE 2-10

memory into buffer register 1. The transmission of the data words is as described in Paragraph 2.3.1.2.

When the last data word is transmitted, a Go/No-Go code word is automatically sent out. The local processor sends back an acknowledge control word as before over the Type 2 input channel. A status word is formed and stored in memory. The IOP command control goes on to the next operation. No re-transmission of the data messages is made in this mode.

2.3.2. Input Data From Local Processor

As in the output data modes, a control word is accessed from memory. In this case, bits 20, 27, and 28 form the code 110. The control word is formed as described in Paragraph 2.3.1.

Each computer has a Type 2 data bus to each local processor that is independent of the other computer buses. This permits multiple paths for transferring data from the local processor to the computer system. Before any data are used, each computer compares the data received over its own bus with the data received over other computer data buses. The data are used according to the results of this comparison. So that all computers may have copies of the data received by the other buses, each computer will automatically send out on the Type 1 output channel any data received over the Type 2 input channel.

Proper storage of these data sets requires that the operand program counters of the Type 1 input channels be preset to the proper addresses. After forming the control word, the IOP command control increments

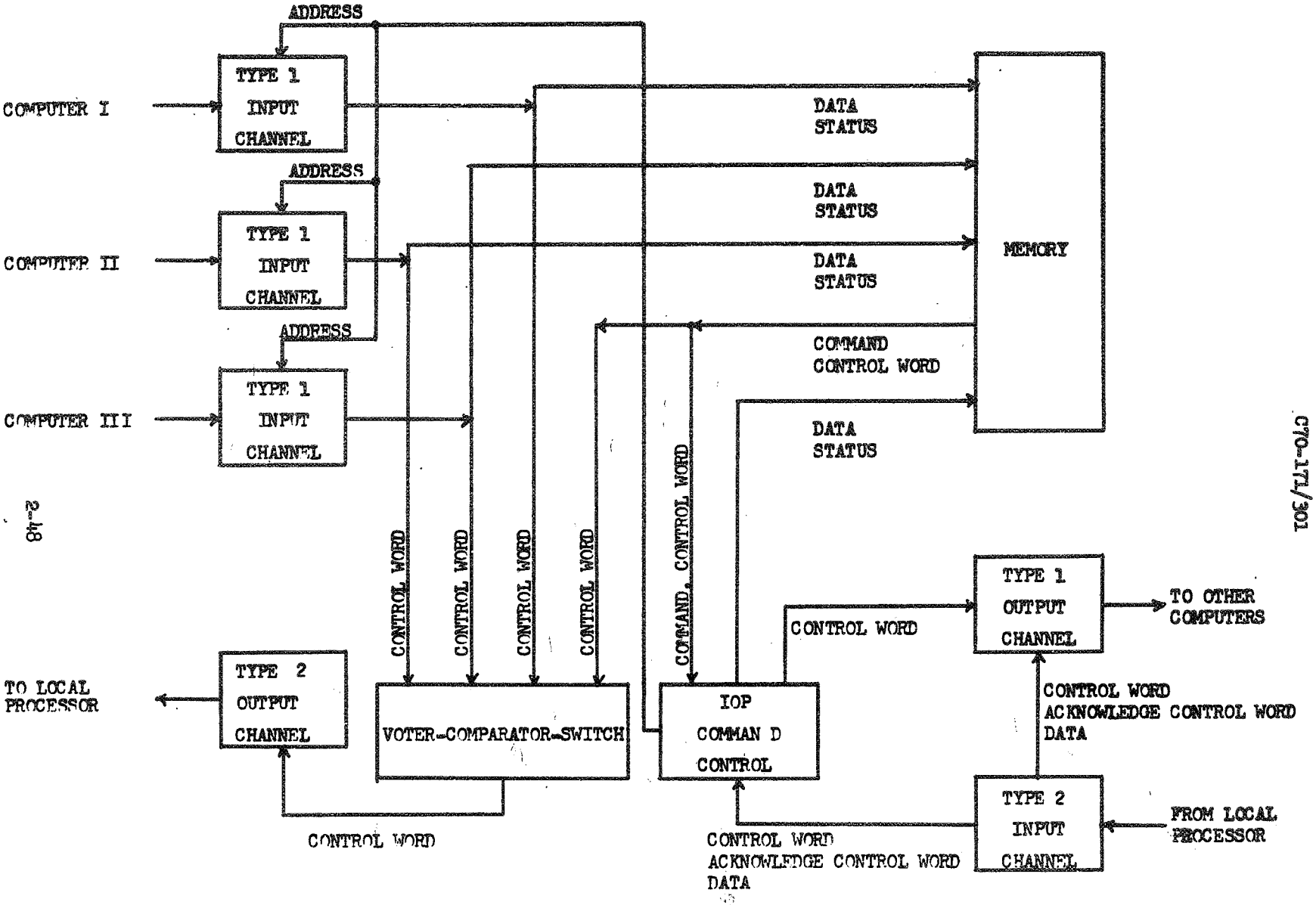
the operand program counter by one. This memory location holds the address for the storage of the data message received by another computer. The succeeding memory locations hold other addresses for the storage of the other data messages. Up to four (4) addresses are stored depending upon the number of data message duplicates desired.

The IOP command control looks at bits 7 through 10 of the control word register. These bits are the reply buses field. Each bit position corresponds to one computer in the system. A binary one in the bit position of the field means that a data message will be received by that computer represented by the bit position. The IOP command control scans the field starting at bit 7. If this position is a one, the memory is accessed, the contents read into the appropriate type 1 input channel operand program counter and the IOP command control operand program counter incremented by one. If the bit in the field is a zero, the IOP command control goes to the next bit position in the field. This is repeated until all four bit positions in the field have been checked and the type 1 input channels set up for the operation.

2.3.2.1 Voting Using Internal VCS

Data flow through the IOP during this operation is diagrammed in Figure 2-12. The operation makes use of the type 2 input and output channels, the type 1 output channel, the voter-comparator-switch, the IOP command control and one to three type 1 input channels.

The control word is sent to the VCS buffer registers and the comparison register of the type 2 input channel. The other control words are received by the type 1 input channels and transferred to the VCS. The control words are voted upon and the majority result sent out



INPUT DATA FROM LOCAL PROCESSOR USING INTERNAL VCS

FIGURE 2-12

C70-171/301

over the type 2 output channel. The results of the voting are stored in status indicators. The control word is received back over the type 2 input channel and compared with the contents of the comparison register. The control word is also sent out over the type 1 output channel. The results of the comparison are stored in status indicators.

The IOP monitors the type 2 input channel for the acknowledge control word. When it arrives, it is checked for proper parity and the LP address and data locations fields are compared with the similar fields stored in the control word register. The results of the comparison are stored in status indicators.

As data are received over the type 2 input channel, the data are stored in memory and sent out over the type 1 output channel. At the same time, data are received from the other computers over the type 1 input channels and stored in memory. The end of the message is signified by the reading of a count of zero by the word counters of the type 1 and 2 input channels. At this point, all status indicators are used to form a status word to be stored in memory.

The decision as to whether the operation has been successful or not is based upon the results of the control word comparison (feedback version against original), failure to find a majority during voting and the results of comparing the two fields of the acknowledge control word and the control word register. Disagreement in the comparisons or failure to find a majority results in a rerun of the operation if the retransmit flag is set. The rerun is made as described in Paragraph 2.3.1.1. Parity errors detected by the type 1 input channels will not cause a retransmission try.

2.3.2.2 Voting Using External VCS

Data flow through the IOP during this operation is diagrammed in Figure 2-13. This operation makes use of the IOP command control, the type 2 input channel, the type 1 output channel and one to three type 1 input channels.

The control word is transmitted out over the type 1 output channel and placed in the comparison register of the type 1 input channel associated with the computer doing the voting. The rest of the operation is the same as that described in Paragraph 2.3.2.1.

2.3.2.3 No Voting

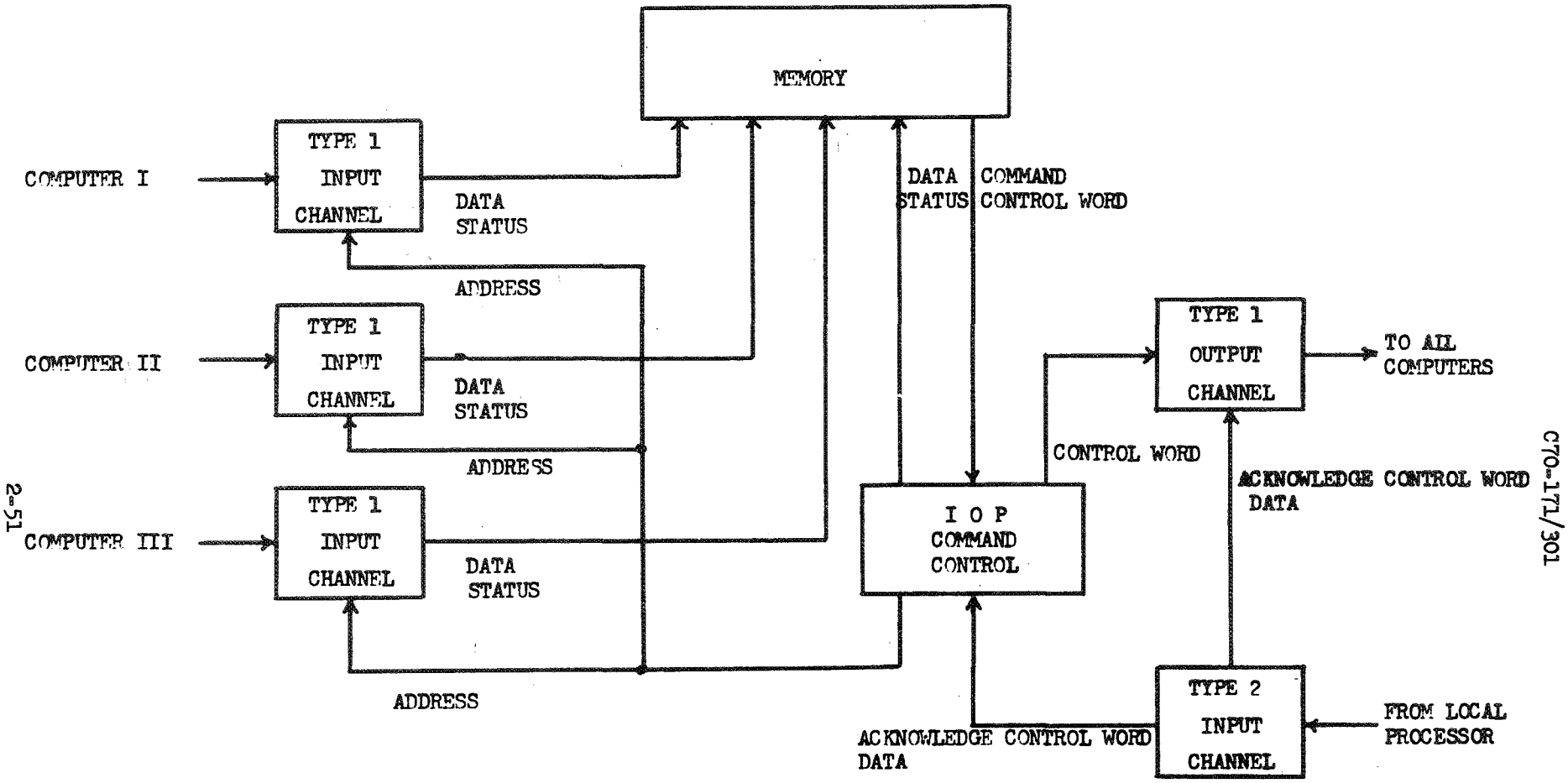
Data flow through the IOP during this operation is diagrammed in Figure 2-14. This operation makes use of the IOP command control and the type 2 input and output channels.

The control word is sent out over the type 2 output channel directly to the local processor. The acknowledge control word is received over the type 2 input channel and checked as in Paragraph 2.3.2.1. The data are received and stored in memory. A status word is formed at the end of the message and stored in memory. The operation will not be repeated regardless of the errors detected in the message.

2.3.3 Request Data From Another Computer

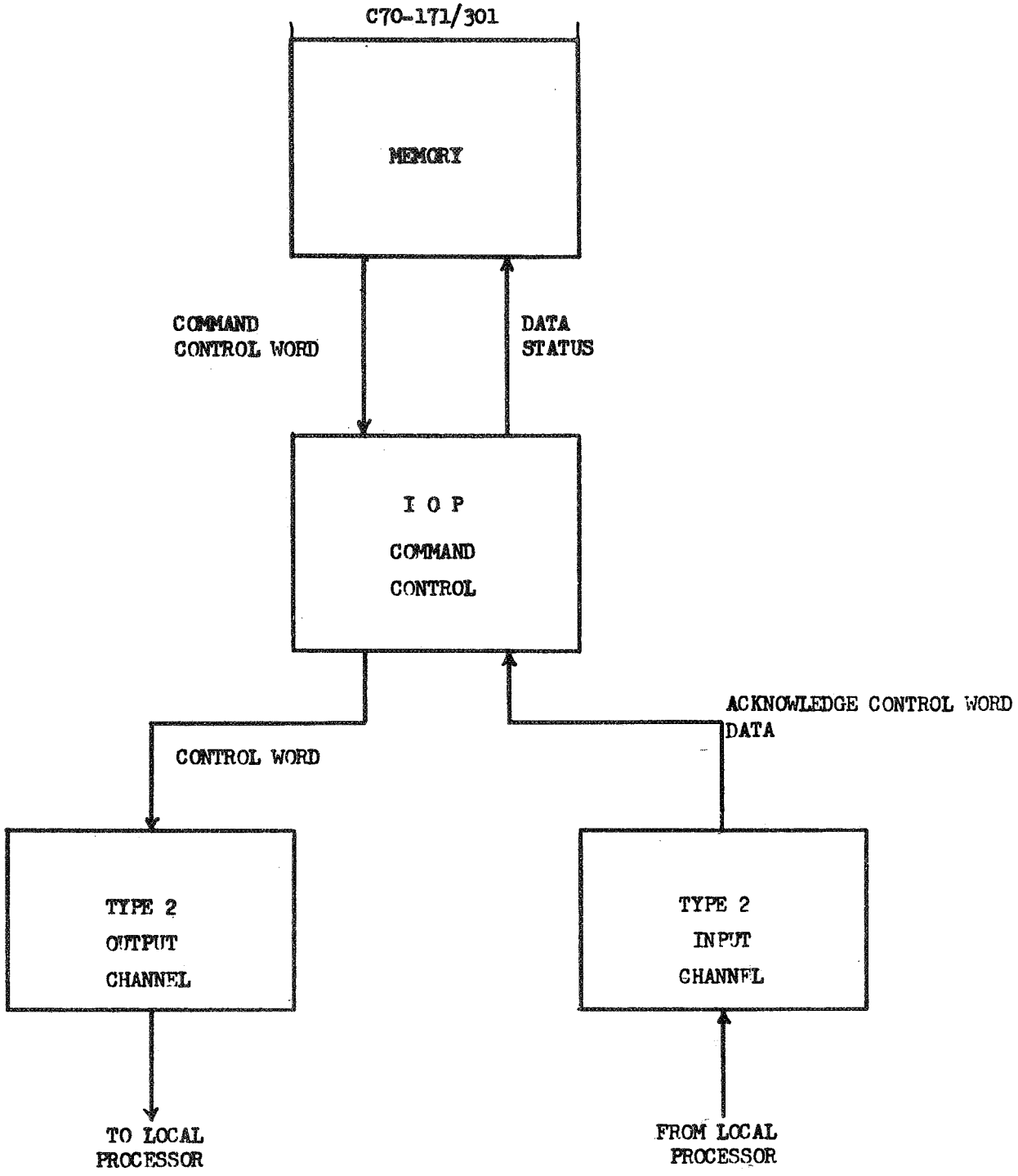
Data flow through the IOP during this operation is diagrammed in Figure 2-15. This operation makes use of the IOP command control and the type 1 output channel.

The control word for this operation is the computer-to-computer (memory) type as identified by the binary code of 111 in bit positions 20, 27 and 28. This control word is made up by filling the computer



INPUT DATA FROM LOCAL PROCESSOR USING EXTERNAL VCS

FIGURE 2-13



INPUT DATA FROM LOCAL PROCESSOR WITHOUT VOTING

FIGURE 2-14

address, memory address and return store address fields with data accessed from memory by using the initial value of the computer address field as an address. The control word has the address of the requesting computer in the return address field. The control word is sent out on the type 1 output channel. The IOP command control goes on to the next operation as the response to this request will be handled automatically by the type 1 input channels.

2.3.4 Input Data From Another Computer

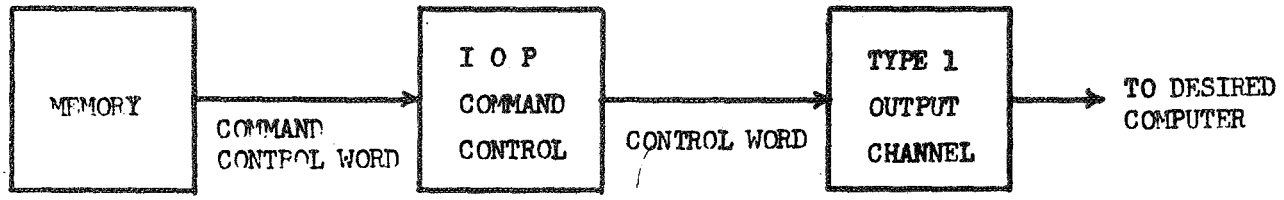
Data flow through the IOP during this operation is diagrammed in Figure 2-16. This operation makes use of one type 1 input channel.

This operation requires no stored commands or any action by the IOP command control. All information necessary is found in the received control word. The control word holds the binary code of 111 in bit positions 20, 27 and 28. The control word is copied into the channel control word register if parity checks on both halves of the control word are correct. As the first data word is received serially into the channel buffer register, the type 1 input channel is set up as follows:

a. The number of words field (bits 1 through 6) of the control word register are copied into the word counter.

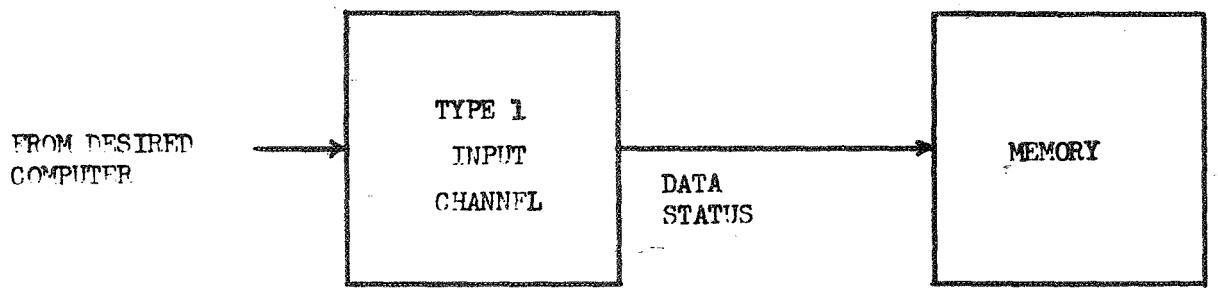
b. The return store address field (bits 7 through 12) of the control word register are copied into the least significant bit positions of the operand program counter. (This limits the area of memory that can be stored into directly by the external computer.)

A bad parity check on either half of the control word causes the type 1 input channel to ignore the rest of the inputs on the bus.



REQUEST DATA FROM ANOTHER COMPUTER

FIGURE 2-15



INPUT DATA FROM ANOTHER COMPUTER

FIGURE 2-16

Each data word received on the data bus is checked for parity. The word counter is decremented one count for every 17 bits received on the bus. The parity bits are discarded and the data bits assembled into 32 bit words and stored in memory. When the word counter reaches zero, a status word is formed and stored in memory after the last data word. The type 1 input channel goes to the idle mode ready for the next operation.

2.3.5 Output Data To Another Computer

There are two ways to initiate a data output to another computer operation. One way is to fetch a control word from memory and the other way is to receive a control word requesting data over the type 1 input channel.

2.3.5.1 Output Requested Data to Another Computer

Data flow through the IOP for the operation is diagrammed in Figure 2-17. This operation makes use of the IOP command control, the type 1 output channel and one type 1 input channel.

The operation starts when the type 1 input channel receives a control word containing the binary code 011 in bit positions 20, 27 and 28. The input channel places the control word in its control word register and sets the type 1 input channel interrupt true.

The IOP command control examines the type 1 input channel interrupts after the completion of each operation. If one is found to be true, the IOP command control proceeds to read from the control word register of that type 1 input channel into buffer register 2 the following:

a. Bits 1 through 6 into bits 1 through 6 (Number of words).

b. Bits 13 through 16 into bits 29 through 32 (Return address).

c. Bits 7 through 12 into bits 21 through 26 (Return store address).

In addition, other bits of buffer register 2 are set as follows:

a. Bits 20, 27, and 28 to binary ones.

b. Bits 7 through 16 to binary zeroes.

This completes the formation of the control word to lead the data message to be sent.

The IOP command control also reads from the channel control word register:

a. The number of words field into the word counter.

b. The memory address field into the least significant bit positions of the operand program counter.

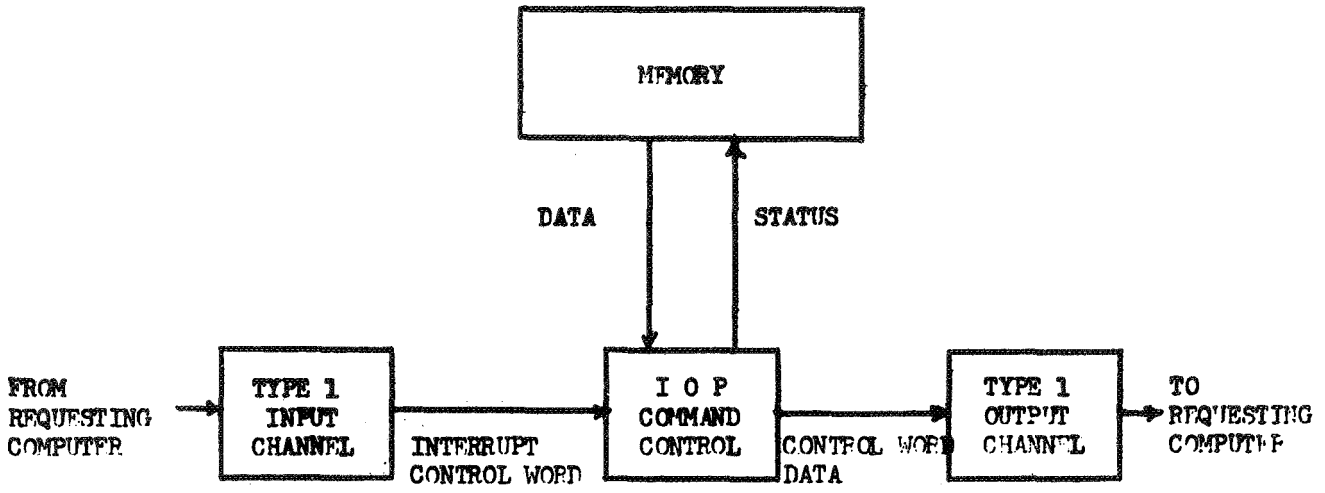
The IOP command control sets the type 1 input channel interrupt freeing the input channel for further operations. The control word in buffer register 2 is sent out over the type 1 output channel and the first data word accessed from memory.

As each data word is transmitted out, the next data word is read from memory so there is no break in the transmission. A parity bit is added for each 16 bits of data transmitted. When the word counter reaches zero, the operation is considered completed.

2.3.5.2 Output Data to Another Computer

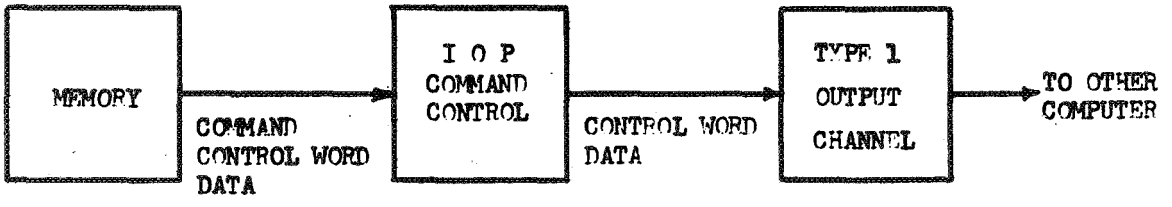
Data flow-through the IOP for this operation is diagrammed in Figure 2-18. This operation makes use of the IOP command control and the type 1 output channel.

The operation begins when a control word having the code 011 in bit positions 20, 27 and 28 is read from memory into the control word



OUTPUT REQUESTED DATA TO ANOTHER COMPUTER

FIGURE 2-17



OUTPUT DATA TO ANOTHER COMPUTER

FIGURE 2-18

register of the IOP command control. The control word is modified by filling the computer address and memory address fields with data accessed from memory by using the initial value of the computer address field as a memory address. The control word is read from the control word register into buffer register 2 and bit position 20 of the buffer register is set to a binary one. The number of words field of the control word register is read into the word counter. The contents of the memory location following that location holding the control word are read into the operand program counter as the address of the first data word in memory. The rest of the operation is the same as that described in Paragraph 2.3.5.1.

2.3.6 Set Matrices In Other Computers

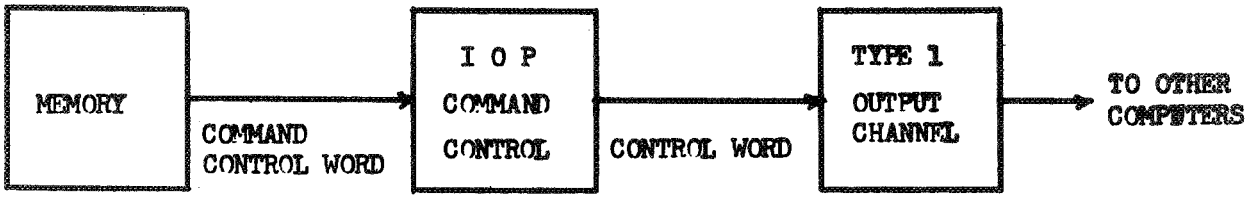
Data flow through the IOP during this operation is diagrammed in Figure 2-19. This operation makes use of the IOP command control and the type 1 output channel.

The operation begins when the control word read from memory into the control word register contains the binary code 0001 in bit positions 25 through 28. The data address field of the control word is read into the least significant bit positions of the operand program counter and the memory location with this address read out. Bit positions 5 through 16 of the control word register are filled with this data. These bits now represent the data that are to be set into the matrices of the other computers.

The contents of the control word register are read into buffer register 2 and then sent out on the type 1 output channel. The transmission of the last parity bit completes the operation.

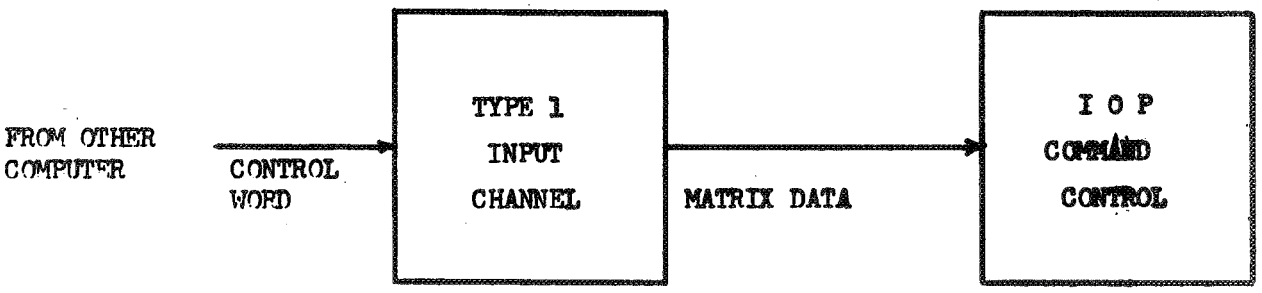
2.3.7 Set Own Matrices According to Data From Another Computer

Data flow through the IOP during this operation is diagrammed in Figure 2-20. This operation makes use of the IOP command control and one type 1 input channel.



SET MATRICES IN OTHER COMPUTERS

FIGURE 2-19



SET OWN MATRICES ACCORDING TO DATA FROM ANOTHER COMPUTER

FIGURE 2-20

The operation is started when the type 1 input channel receives a control word containing the binary code 0001 in bit positions 25 through 28. Bit positions 23 and 24 are investigated to determine which matrix or matrices are to be set. The data in bit position 5 through 16 are transferred to the appropriate matrices as specified by bits 23 and 24. The type 1 input channel returns to the idle mode completing the operation.

2.3.8 Sample Matrices In Another Computer

Data flow through the IOP during this operation is diagrammed in Figure 2-21. This operation involves the IOP command control, the type 1 output channel and one type 1 input channel.

The operation starts when a control word containing the binary code 1001 in bits 25 through 28 is read from memory into the control word register of the IOP command control. The computer address field is set up as before and the control word sent out over the type 1 output channel. The IOP command control sets the word counter to 34. The word counter is counted down at the logic clock rate of 1 megahertz. During this 34 microsecond period, the IOP command control monitors the type 1 input channel interrupts. When the correct interrupt is set true, the IOP command control compares bits 21 through 26 of the channel control word register with bits 7 through 12 of the IOP command control word register. If they match, the IOP command control resets the word counter and goes on to the next operation.

If the two fields do not match, the IOP command control resumes monitoring the interrupts. If the word counter reaches zero before the right control word is received, the IOP forms a status word indicating the failure to receive the requested data and stores it away in memory. The IOP command control goes on to the next operation.

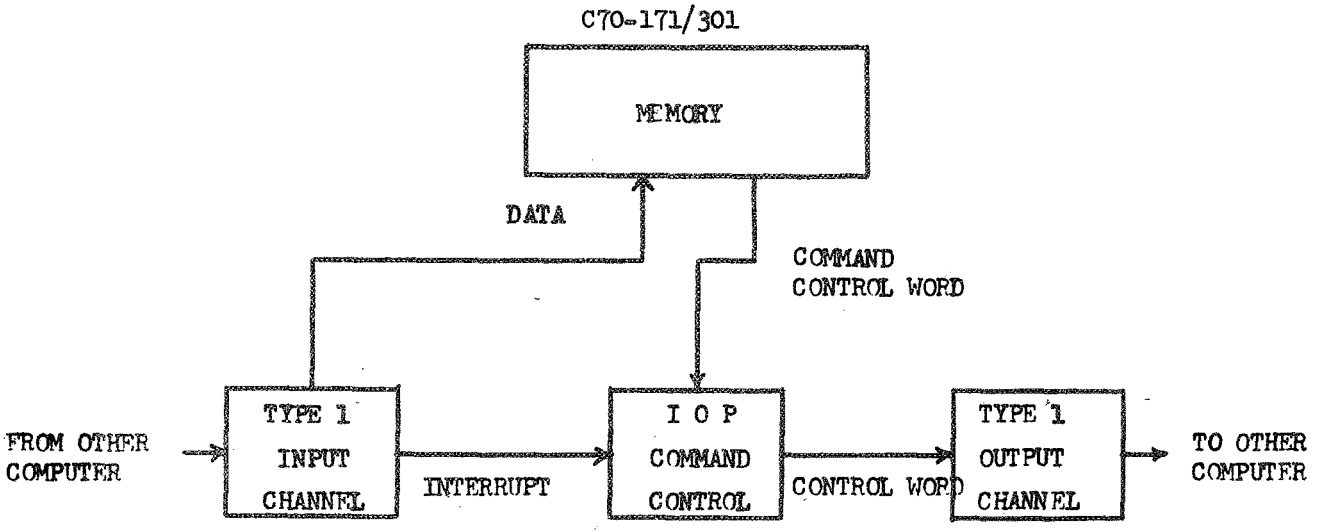
2.3.9 Sample Matrices as Requested by Another Computer

Data flow through the IOP during this operation is diagrammed in Figure 2-22. This operation makes use of the IOP command control, the Type 1 output channel, and one to three Type 1 input channels.

The operation starts when a control word with the binary code 1001 in bit positions 25 through 28 is received by one Type 1 input channel. The receiving channel sets its interrupt true. The IOP command control samples the interrupts after completion of its last operation and, upon finding one true, proceeds to check the bits 25 through 28 to determine the type of operation.

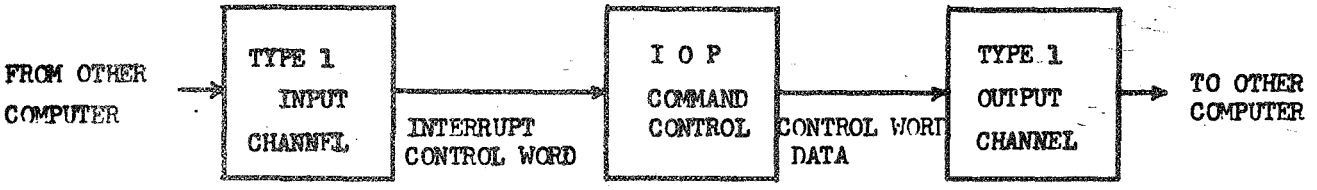
The IOP command control then reads the return store address field into bits 21 through 26 of buffer register 2. Bits 23 and 24 of the channel control word register are read into the control word register of the IOP command control word register. Bits 20, 27 and 28 are set to ones. If bits 23 and 24 of the control word register are set to zeroes, then bits 6 through 1 of the control word register are set to 000011 respectively. Any other code in bits 23 and 24 causes 000001 to be set into bit positions 6 through 1 respectively. These bits are the number of words field and are set to a value of one if only one matrix is to be sampled or to a value of three if all matrices are to be sampled.

The computer address field (bits 29 through 32) is generated according to which computers requested the sampling. The IOP command control test each Type 1 input channel interrupt. If the interrupt is true, the bit positions 25 through 28 of the channel control word register are tested for the code 1001. If the code is there, a one is placed in the computer address field bit position corresponding to that computer. If the code is not there or the interrupt was false, a zero is placed in the address field. This is done for all



SAMPLE MATRICES IN ANOTHER COMPUTER

FIGURE 2-21



SAMPLE MATRICES AS REQUESTED BY ANOTHER COMPUTER

FIGURE 2-22

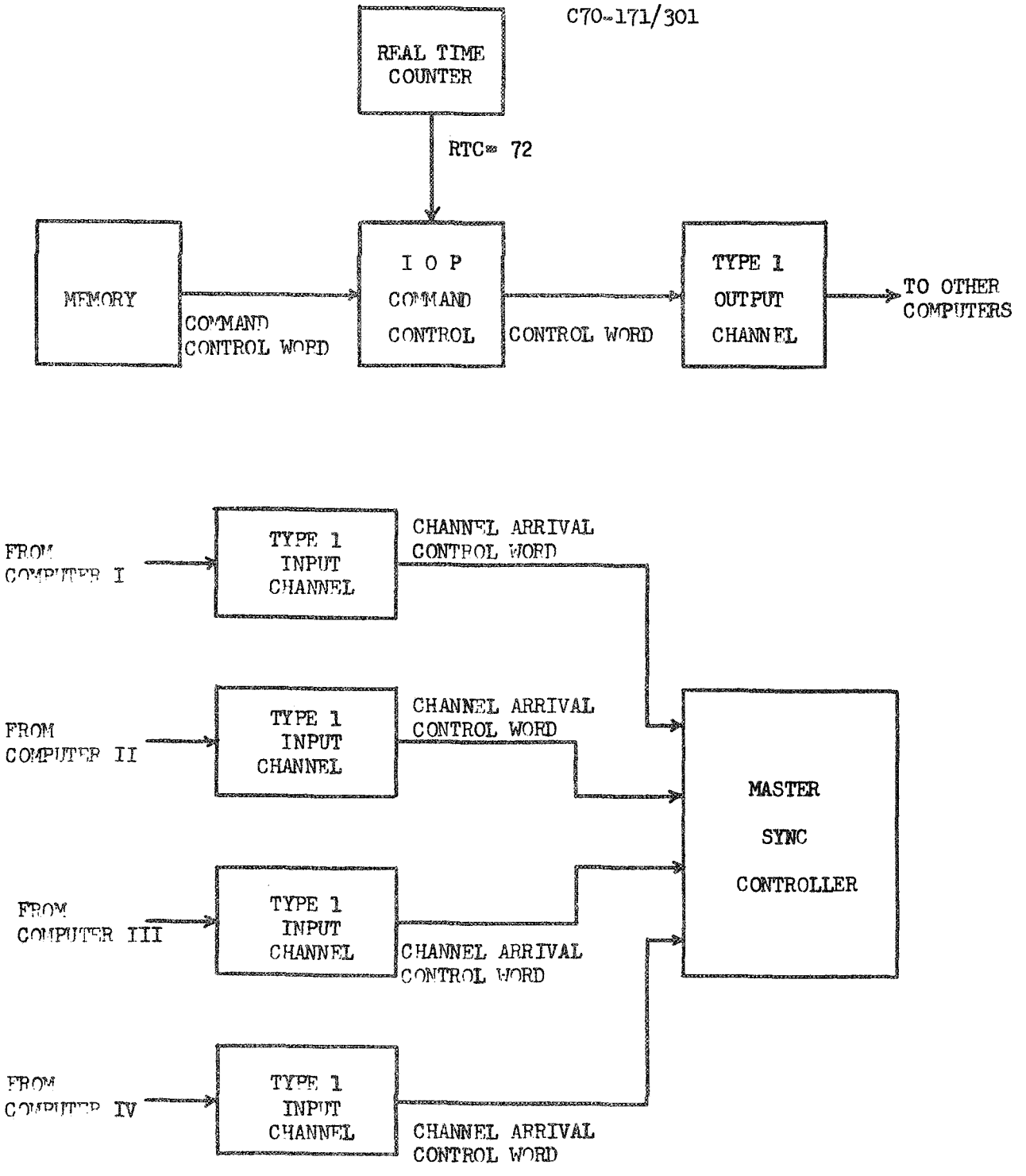
the type 1 input channels. The control word is now complete and is transferred to buffer register 2.

The control word is transmitted out over the type 1 output channel. After the first half of the control word is transmitted, either the R or specified matrix is read into the lower half of buffer register 2. The R matrix is read in if all matrices are to be sampled. If only one matrix was to be sampled, the operation ends after the data are transmitted from buffer register 2. If all matrices are to be sampled, the P and S matrices are copied into buffer register 1. The operation is complete when all the data in both buffer register 1 and 2 are transmitted.

2.3.10 Master Sync Operation

The master sync operation is started when the IOP executes the halt and proceed command with the flag set true. The master sync controller (MSC) mode control goes from the idle mode to a mode waiting for the arrival of the first master sync control word. The execution of the flagged command causes the IOP control to idle until the real time counter reaches a count of 72. At this point, the IOP accesses a dedicated memory location for the master sync control word. This control word is sent to all computers over the type 1 output channel. The start select flip flop in the IOP command control is set so the next IOP program cycle will be started when the cycle start discrete of the MSC goes true rather than when the real time counter interrupt goes true.

The control word is received by the type 1 input channel of the computer and bit positions 26, 27 and 28 are investigated. If these bits form the binary code 101, the type 1 input channel sets its channel arrival discrete true. The control word is held in the channel buffer register so that the MSC can read the return address code.



MASTER SYNC OPERATION

FIGURE 2-23

In the MSC, the channel arrival discrete sets a channel arrival flip flop. At the next clock time, the state of a binary counter (counter 1) is read into an arrival time register. At the same time, bits 13 through 16 of the channel buffer register are copied into the master sync word address register. The type 1 input channel is released for other uses. The MSC has one set of registers for each type 1 input channel. The above procedure is repeated for each master sync control word received by the computer before the MSC completes the master sync operation.

The first master sync control word that arrives causes the start counter 1 flip flop to be set. This stores the fact that one control word has arrived and starts the timing of the intervals between the various control word arrivals. Counter 1 provides this timing by counting at the logic clock rate of one megahertz. The start counter 1 flip flop being set enables the next channel arrival discrete to set the start counter 2 flip flop. Counter 2 also counting at logic clock rate overflows after eight counts (eight microseconds) setting the start cycle flip flop. One clock time later, the cycle start flip flop is reset. The output of the cycle start flip flop appears as a discrete that goes true for one microsecond that is used to initialize and start the real time counter.

2.4 Real Time Counter

The Real Time Counter is used to define a reference time period for the computer. The counter is set to a known value and counted down to zero at a fixed rate. When the counter reaches zero, the IOP and the Central Processing Unit (CPU) are notified by a pulse discrete from the counter. The counter is reset and the operation is repeated.

The RTC is a programmable counter fifteen bits long. The counter is decremented at the computer basic logic clock rate of one megahertz. This results in a maximum timing period of 32,768 microseconds. The counter constant is stored in a dedicated memory location to enable recovery from a power transient.

When the computer is initially turned on or recovers from a power transient, the IOP reads the contents of the dedicated memory location into a reset value store register (RVSR). When the RVSR is loaded, the RVSR contents are automatically read into the RTC. The RTC is now decremented at clock rate. Upon reaching a zero count, the RTC issues a discrete pulse signal to the IOP and the CPU. The contents of the RVSR are read into the counter and the operation repeated.

To maintain synchronism between the computers in the complex, periodically the individual real time counters must be reset according to a master signal. This operation is called the master sync operation. When the master sync operation is performed, the RTC is filled with the contents of the RVSR and the timing started at that point.

2.5 Built-In Test Equipment Counter

The BITE counter is used to detect faults in either the couputer hardware or software during operation. The counter is set to a set value and counted down to zero at some rate. If the counter reaches zero, an interrupt is issued indicating that the computer has malfunctioned. The counter is prevented from reaching zero by having the IOP and the CPU program work together to reset the counter before it reaches zero.

The BITE counter is a programmable counter fifteen bits long. The counter is decremented at the basic computer logic clock rate of one megahertz. This results in a maximum period of 32,768 microseconds. The counter constant is determined by the contents of a dedicated location in memory.

The BITE counter is initially loaded and started after application of power to the computer by the IOP accessing memory and transferring the contents of that location into the counter. The counter is counted down until it reaches a count of 1024. At this point, the BITE counter load request discrete is set true. After completing the execution of every command, the IOP samples the BITE counter load request discrete. If the discrete is false, the IOP continues on to the next command. If the discrete is true, the IOP will access the memory for the constant and load the contents of the location accessed into buffer register 1. Each time the IOP reads this memory location, zeroes are written into the location. This is done to force the CPU program to write into the location before the next counter load request is made or the counter will not be reset correctly. The load counter discrete is set true causing the contents of buffer register 1 to be read into the BITE counter. The BITE counter load request discrete is set false. The counter is now reset and the operation repeats.

2.6 MASTER SYNC CONTROLLER

Periodically during system operation, the IOP programs of the computers must be synchronized to have proper operation during the critical mode. The CPU program shall determine when this master sync operation will occur.

The master sync operation is started when the IOP executes the halt and proceed command with the flag set true. The master sync controller (MSC) mode control goes from the idle mode to a mode waiting for the arrival of the first master sync control word. The execution of the flagged command causes the IOP control to idle until the real time counter reaches a count of 72. At this point, the IOP accesses a dedicated memory location for the master sync control word. This control word is sent to all computers over the type 1 output channel.

The control word is received by the type 1 input channel of the computer and bit positions 26, 27 and 28 are investigated. If these bits form the binary code 101, the type 1 input channel sets its channel arrival discrete true. The control word is held in the channel buffer register so that the MSC can read the return address code.

In the MSC, the channel arrival discrete sets a channel arrival flip flop. At the next clock time the state of a binary counter (counter 1) is read into an arrival time register. At the same time, bits 13 through 16 of the channel buffer register are copied into the master sync word address register. The type 1 input channel is released for other uses. The MSC has one set of registers for each type 1 input channel The above procedure is repeated for

each master sync control word received by the computer before the MSC completes the master sync operation.

The master sync operation is based upon the following:

> The real time counter of the computer will be set to an initial value eight microseconds after the second master sync control word at the computer.

The first master sync control word that arrives causes the start counter 1 flip flop to be set. This stores the fact that one control word has arrived and starts the timing of the intervals between the various control word arrivals. Counter 1 provides this timing by counting at the logic clock rate of one megahertz. The start counter 1 flip flop being set enables the next channel arrival discrete to set the start counter 2 flip flop. Counter 2 also counting at logic clock rate overflows after eight counts (eight microseconds) setting the start cycle flip flop. One clock time later the cycle start flip flop is reset. The output of the cycle start flip flop appears as a discrete that goes true for one microsecond that is used to initialize and start the real time counter.

As a fault isolation aid, the relative times of arrival for the four master sync words are stored along with the addresses of the computers originating them. This information will be held in the registers until the next master sync operation when it will be replaced by the new data.

Further fault isolation aid is given by the states of the all-received and not-all-received flip flops. These indicators tell whether or not the correct number of control words were received during the eight microseconds after the arrival of the second control word. A mask register is provided to mask out any computer that has malfunctioned to eliminate faulty master sync operations.

The master sync control was designed to detect a failed computer that has drifted too far out of sync and also to continue operating correctly if a computer has drifted out of tolerance, but cannot be isolated. This situation can happen quite easily, e.g., if the tolerance is Δt, then as long as all sync words arrive within Δt, everything is functioning properly. However, if the words arrive within 2 Δt, it can be deduced that a failure has occurred, but unless one computer is more than Δt out of sync with <u>each</u> of the other computers, the failure cannot be isolated. This difficulty in failure isolation results since no absolute timing reference is available. To prevent using a failed computer as the synchronizing source, the synchronization occurs on the basis of the second received master sync word.

3.0 BLOCK DIAGRAMS OF MECHANIZATION

The following block diagrams illustrate the major elements needed for proper functioning of the IOP. Associated with each diagram is a table listing these elements and giving the number of bits (flip flops) required to mechanize each element. The tables also give the number of times the subsystem is repeated in the IOP.

TABLE 3-1
BITE AND REAL TIME COUNTER ELEMENTS
(One Set Required Per Computer)

	Function	Bits
1.	Reset Value Store Register	15
2.	Real Time Counter	15
3.	Real Time Counter Interrupt	1
4.	BITE Counter	16
5.	BITE Counter Load Request Discrete	1
6.	Computer No Go Discrete	1
7.	Mode Control	4

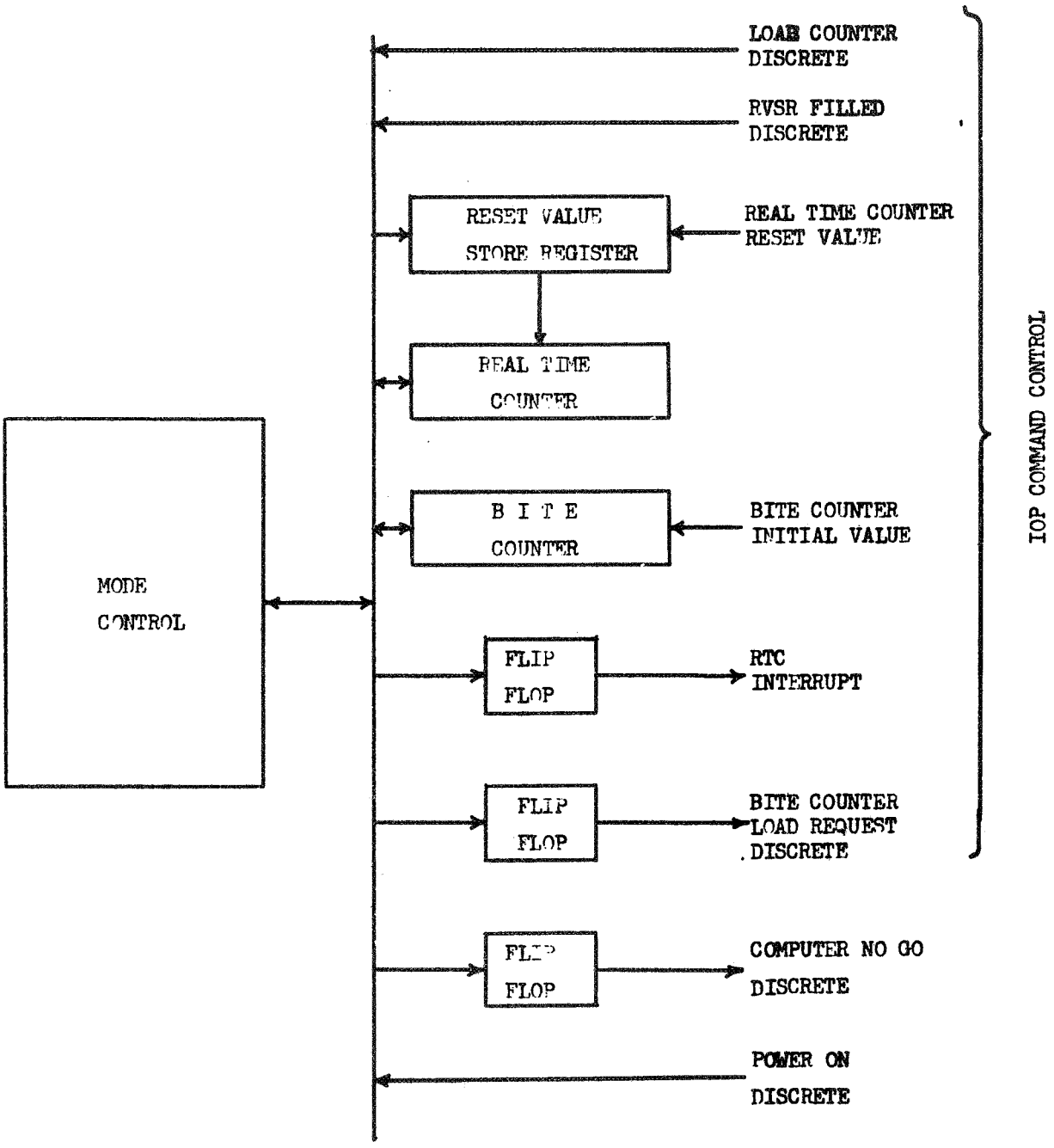


Figure 3-1. BITE AND REAL TIME COUNTERS

TABLE 3-2

MASTER SYNC CONTROLLER ELEMENTS
(One Set Required Per Computer)

	Function	Bits
1.	Counter 1	4
2.	Counter 2	3
3.	Start Counter 1	1
4.	Start Counter 2	1
5.	All Received Indicator	1
6.	Not All Received Indicator	1
7.	Start Cycle Discrete	1
8.	Mask Register	4
9.	Mode Control	5
10.	Input Channel (4 required per Master Sync Controller)	
	a. Master Sync Word Address Register	4
	b. Arrival Time Register	4
	c. Channel Arrival Indicator	1

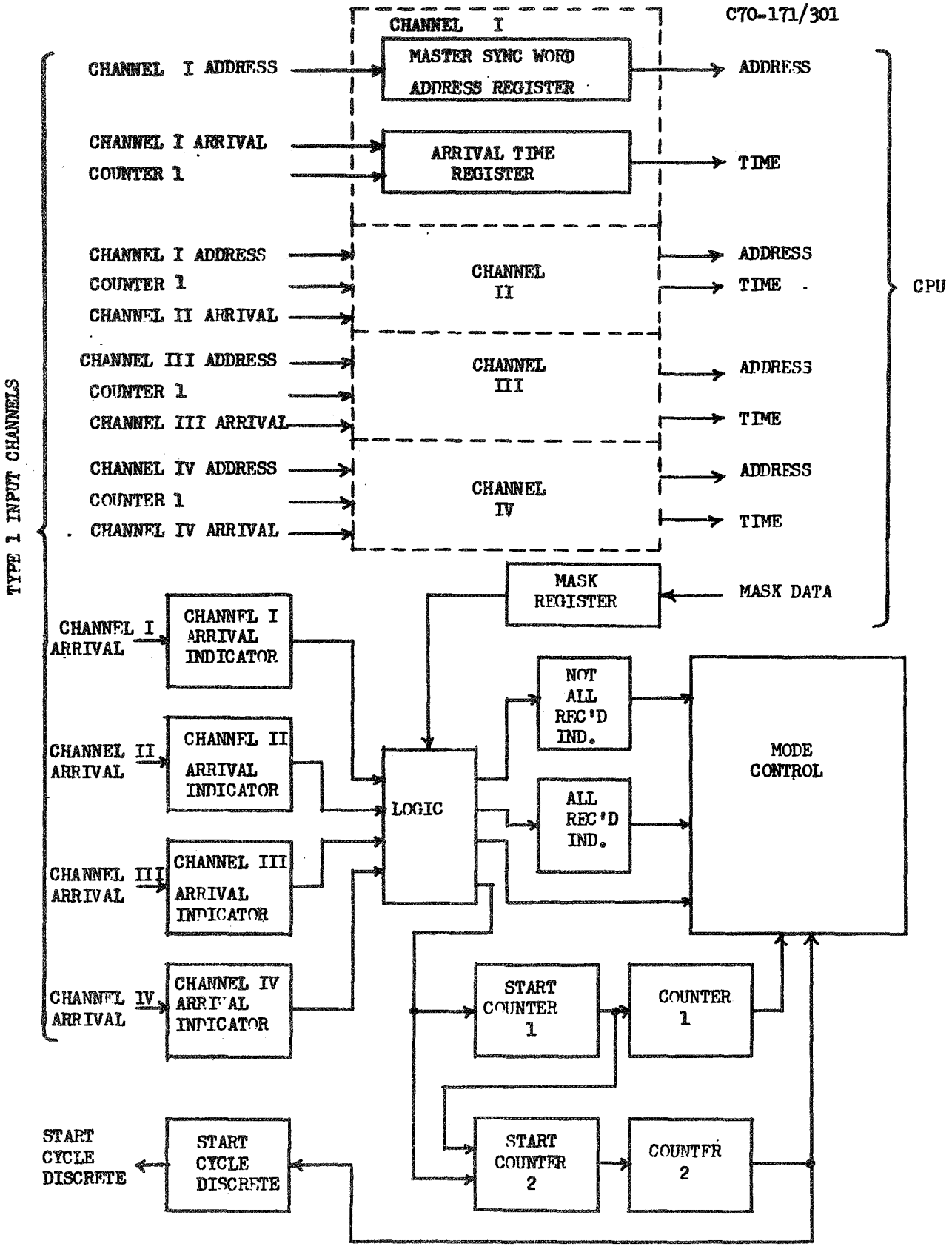


Figure 3-2. MASTER SYNC CONTROLLER

TABLE 3-3

TYPE I INPUT CHANNEL ELEMENTS

(Four Sets Required Per Computer)

	Function	Bits
1.	Bit Counter	6
2.	Word Counter	6
3.	Operand Program Counter	16
4.	Channel Buffer Register	34
5.	Parity Check A	1
6.	Parity Check B	1
7.	Parity Error Indicator	1
8.	Channel Control Word Register	34
9.	Type 1 Input Channel Interrupt	1
10.	Input Buffer Register	32
11.	Memory Write Discrete	1
12.	Comparison Register	80
13.	Comparison Indicator	2
14.	Comparison Counter	3
15.	Channel Arrival Discrete	1
16.	Operation Complete Discrete	1
17.	Mode Control	7

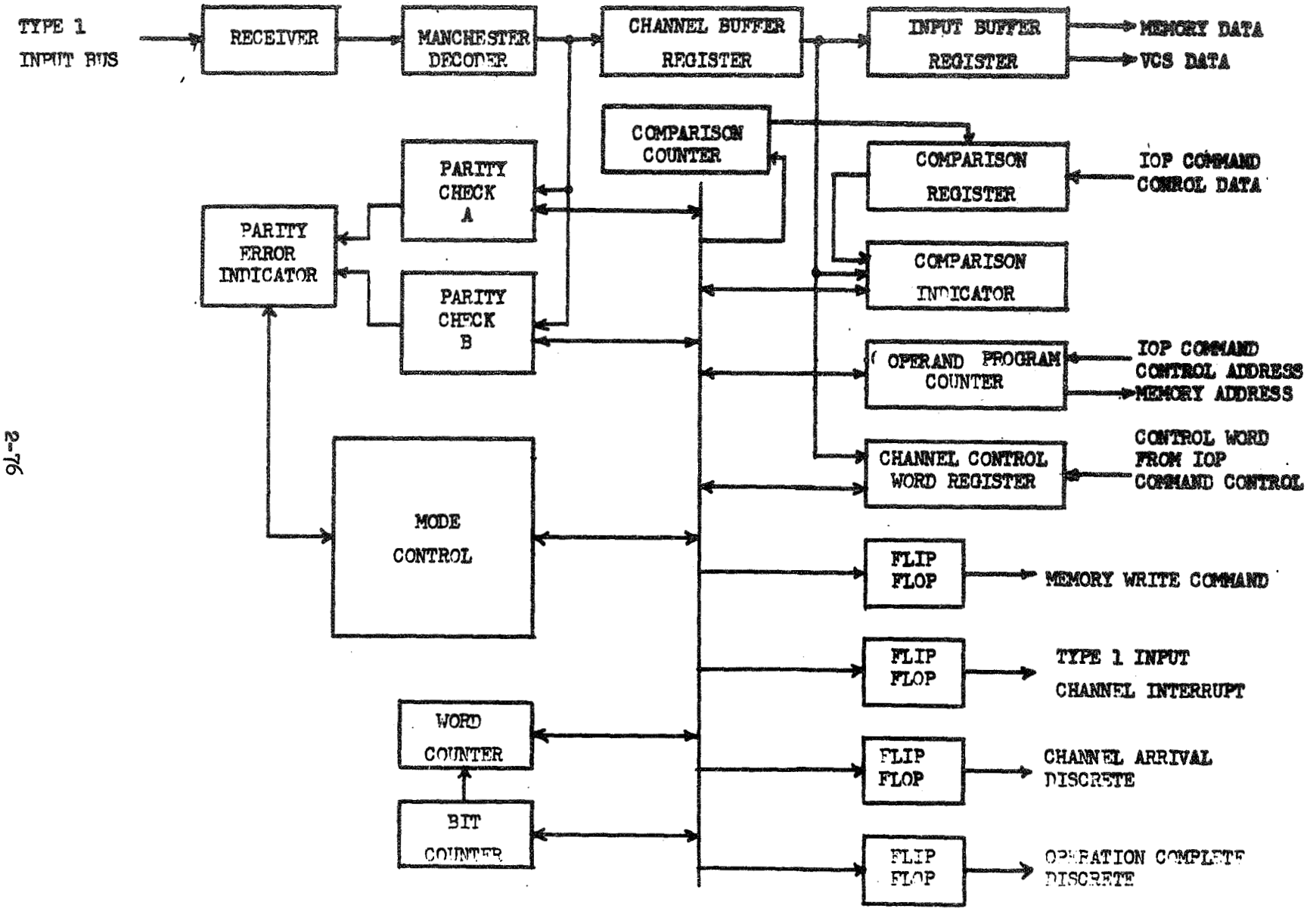


Figure 3-3. TYPE 1 INPUT CHANNEL

TABLE 3-4

TYPE 2 INPUT CHANNEL ELEMENTS
(One Set Required Per Computer)

	Function	Bits
1.	Bit Counter	6
2.	Parity Check	1
3.	Comparison Register	80
4.	Word Counter	6
5.	Signal Detector	1
6.	Type 2 Input Channel Operating Indicator	1
7.	Comparison Indicator	1
8.	Type 2 Input Channel Operation Complete Indicator	1
9.	Mode Control (Including Parity Error Indicator)	7

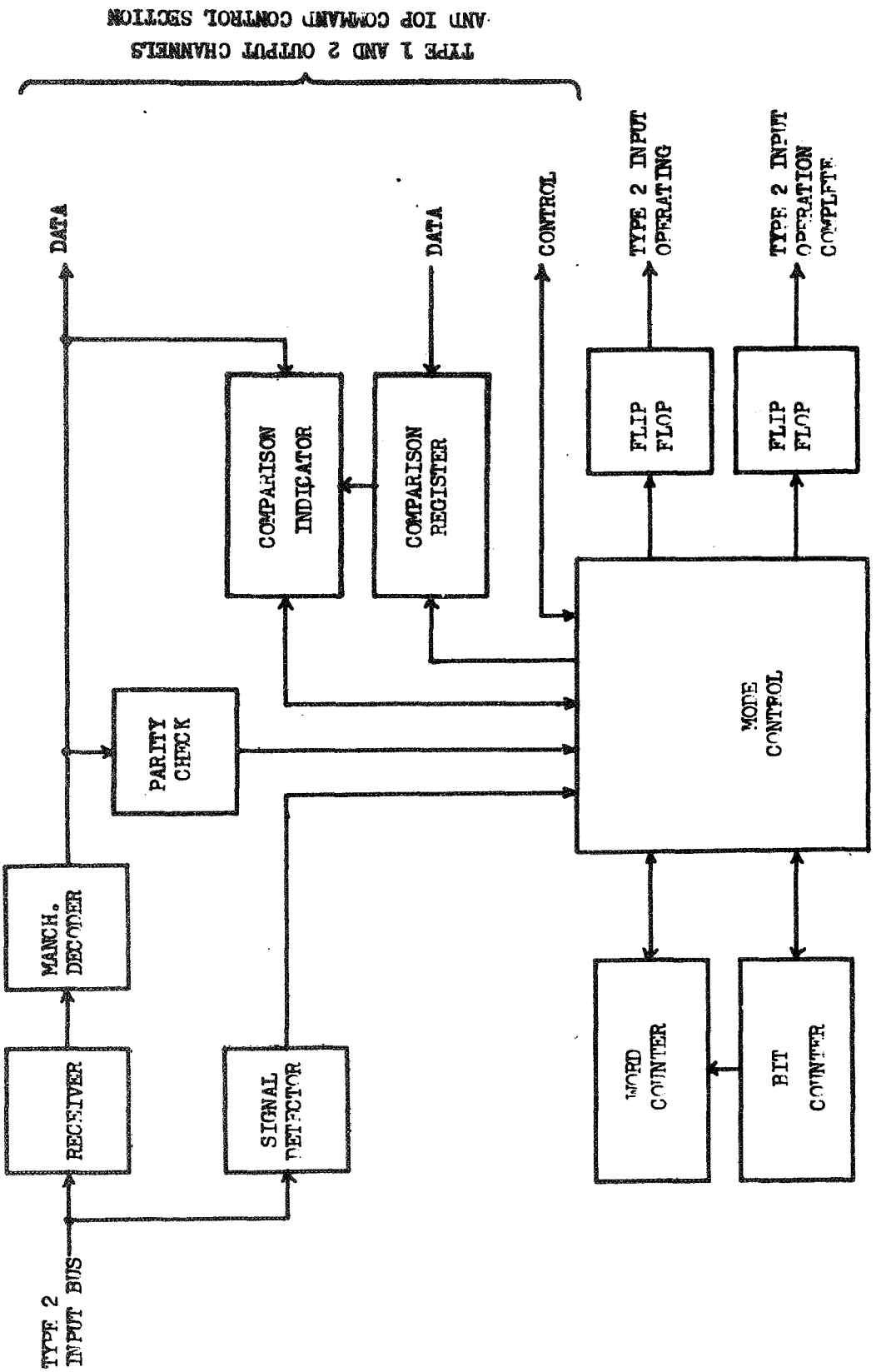


Figure 3-4. TYPE 2 INPUT CHANNEL

TABLE 3-5

TYPE 1 AND 2 OUTPUT CHANNELS AND IOP COMMAND CONTROL ELEMENTS

(One Set Required Per Computer)

	Function	Bits
1.	Command Program Counter	16
2.	Operand Program Counter	16
3.	Reset Value Store Register Loaded Indicator	1
4.	Command Store Register	32
5.	Interrupt Counter	2
6.	Buffer Register 1	32
7.	Buffer Register 2	32
8.	Word Counter	6
9.	Bit Counter	6
10.	Parity Generator	1
11.	R Matrix	16
12.	P Matrix	16
13.	S Matrix	16
14.	Critical Indicator	1
15.	Control Word Register	32
16.	VCS Switched Indicator	1
17.	Input/Output Indicator	1
18.	Retransmission Indicator	1
19.	Retransmit Counter	2
20.	Acknowledge Word Store	16
21.	Acknowledge Word Error Indicator	1
22.	Acknowledge Word Indicator	2
23.	Overflow Indicator	1
24.	BITE Counter Filled Indicator	1
25.	Sync Error Indicator	1
26.	Real Time Counter Error Indicator	1
27.	Start Select Indicator	1
28.	Master Sync Late Indicator	1
29.	Own VCS Indicator	1
30.	Mode Control	9

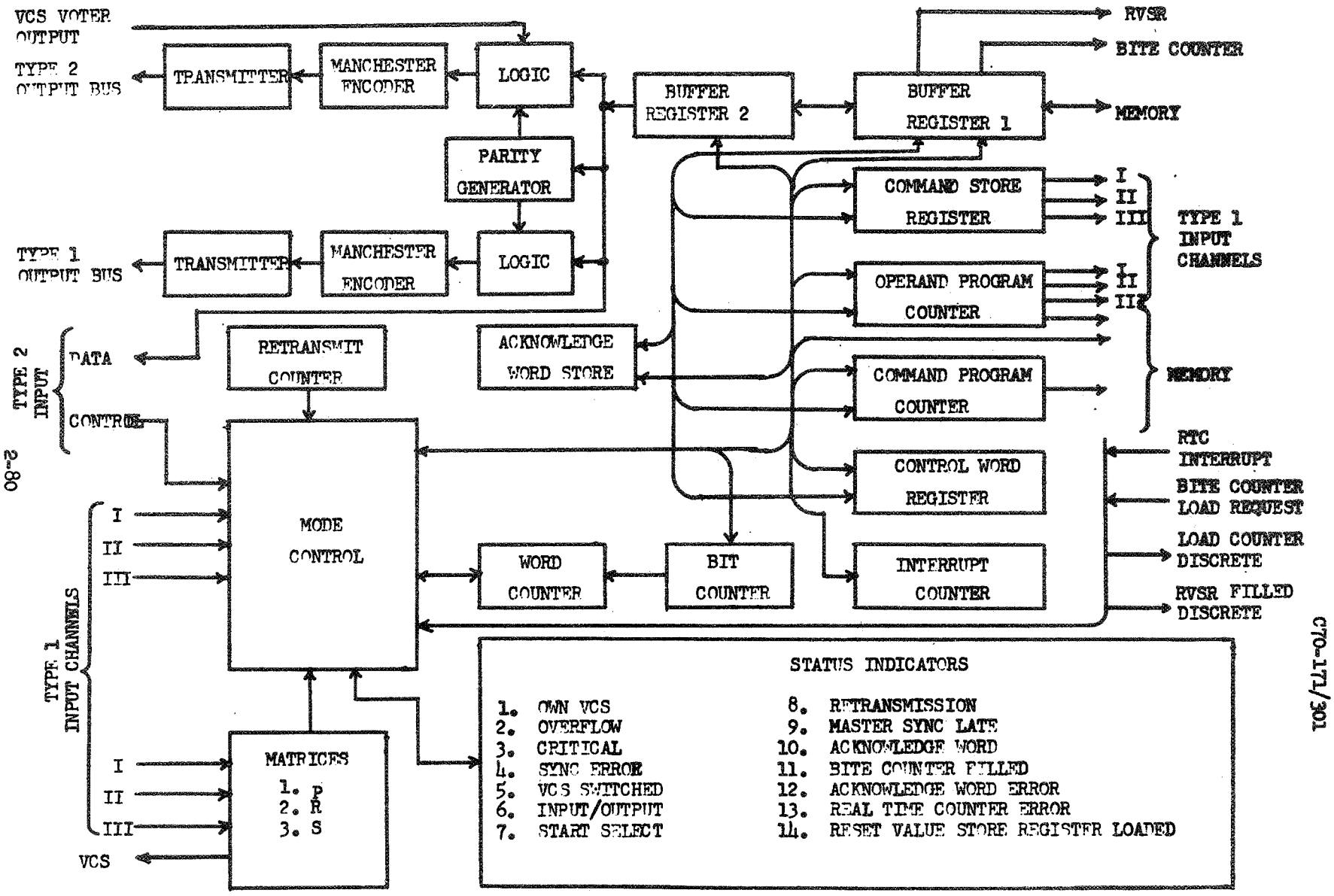


Figure 3-5. TYPE 1 AND 2 OUTPUT CHANNELS AND IOP COMMAND CONTROL

TABLE 3-6

VOTER-COMPARATOR-SWITCH ELEMENTS

(One Set Required Per Computer)

	Function	Bits
1.	VCS Advance Register Discrete	1
2.	VCS Bit Counter	5
3.	Voter	1
4.	VCS Buffer Register 4	17
5.	Agree/Disagree Indicator	1
6.	Transmitter On/Off Indicator	1
7.	Voter Error Indicators	5
8.	I/O Store	1
9.	VCS Word Counter	6
10.	VCS Operation Complete Discrete	1
11.	Mode Control A	5
12.	VCS Input Channel (4 required per VCS)	
	a. VCS Buffer Register 1	17
	b. VCS Buffer Register 2	17
	c. VCS Buffer Register 3	17
	d. VCS Buffer Register 1 Indicator	1
	e. VCS Buffer Register 2 Indicator	1
	f. VCS Buffer Register 3 Indicator	1
	g. Mode Control B	4

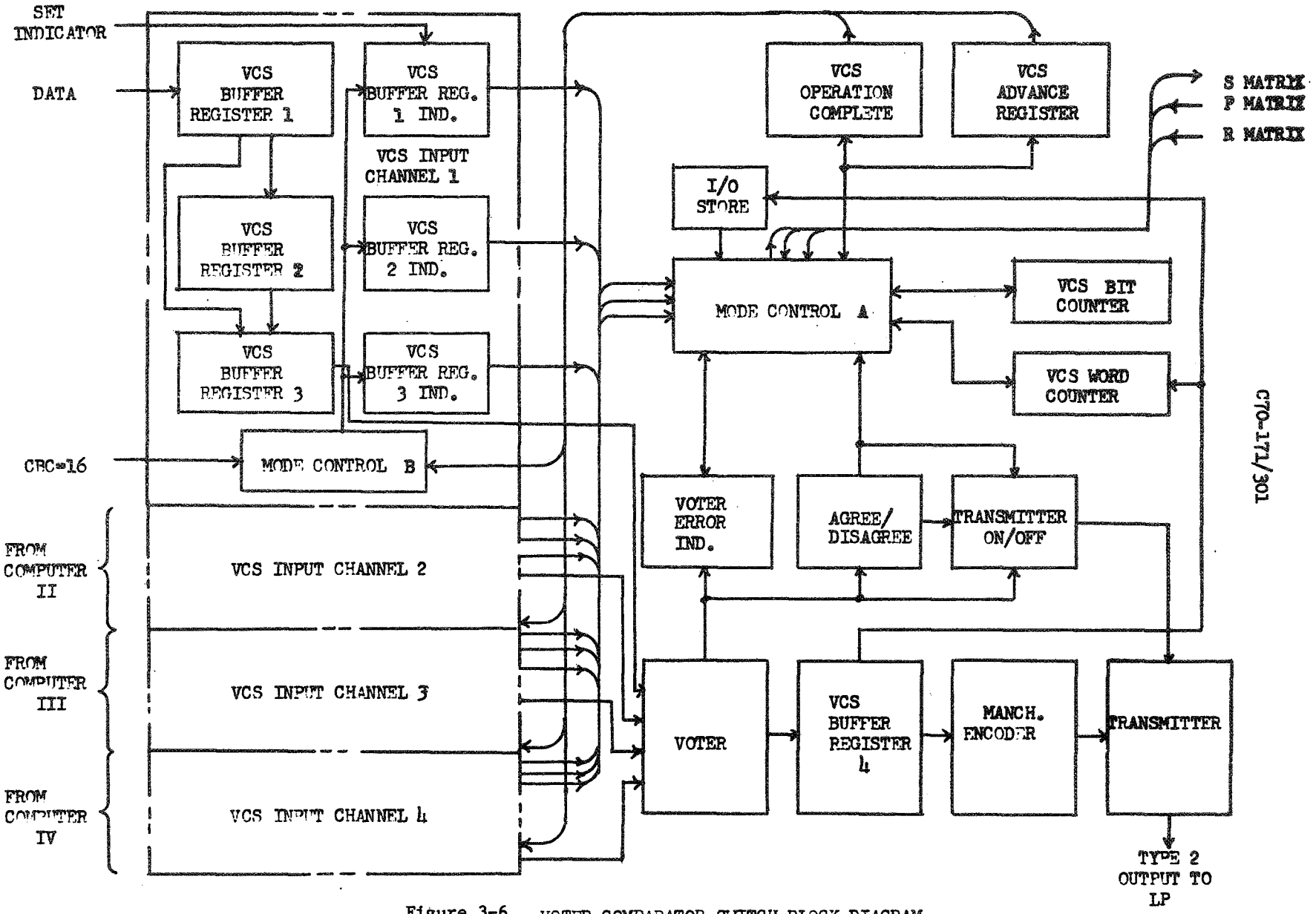


Figure 3-6. VOTER-COMPARATOR-SWITCH BLOCK DIAGRAM

4.0 OPERATIONAL FLOW CHARTS

A link between circuit operation description and the actual hardware elements is the operational flow chart. In this chart, the operations are broken down into the lowest element of actions performed during each clock time. By doing this, one can determine the circuit elements required, the relationships of these elements during operation and the time to perform the operation.

The operations to be performed by the IOP have been described in Section 2. In this section, the details of these operations are shown by means of the flow charts. A by-product of flow charting is the separating out of the independent functional areas of the IOP. These were found to be:

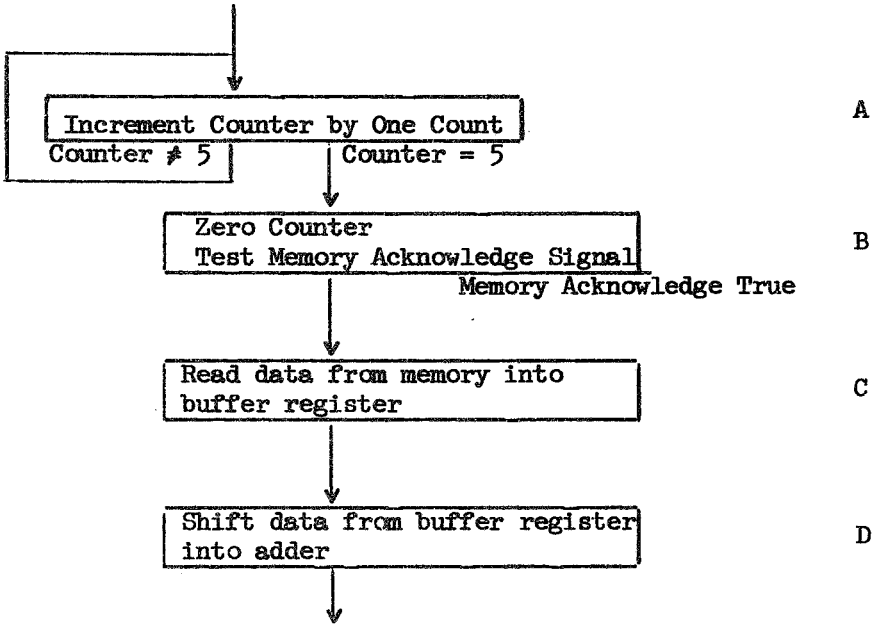
1. Real Time Counter
2. BITE Counter
3. Master Sync Controller
4. Type 1 Input Channel
5. Type 2 Input Channel
6. Type 1 and Type 2 Output Channels and IOP Command Control
7. VCS Input Channel
8. VCS Voter and Output Channel

These areas are not completely independent of each other but require a small amount of information from the other parts to carry out their own operations. The flow charts also point out this interplay between the functional areas and show the amount and timing of the interplay.

4.1 General

The flow charts are organized to show the clock time by clock time operations. All capabilities of each functional area are shown in the charts. In reading the charts, one should make reference to the block diagrams of the IOP and the command and control word descriptions given in sections 2.2 and 3.

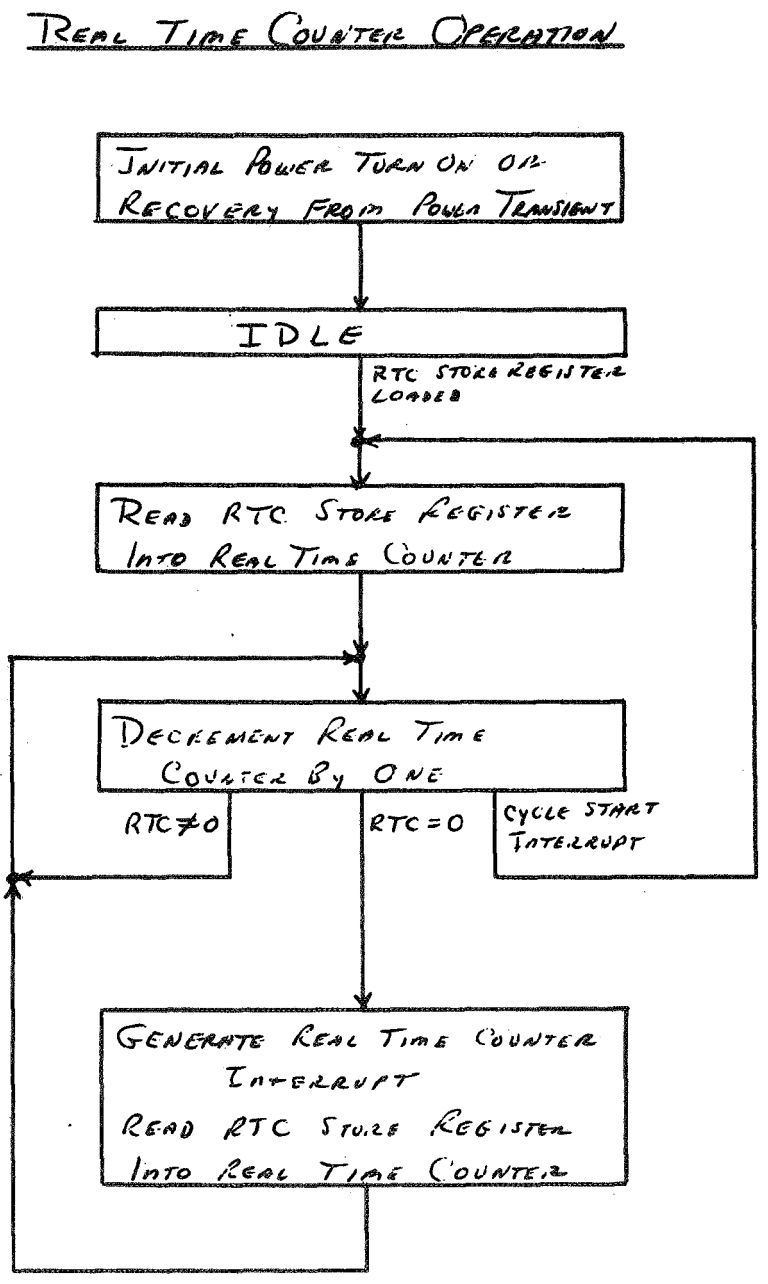
The charts are largely self-explanatory. Each box in the chart describes all the actions that occur during a single clock time. In most cases, exit is made from the box at the end of the clock time. In the event that there are multiple exit paths, each path is shown along with the condition necessary to take the path. In some instances, exit from a box is dependent upon a condition determined by another functional area such as the state of the memory acknowledge signal from the memory. The circuit may stay in the box until the condition is met at the same time a clock time ends. This is the only situation where multiple time periods are spent in a box. A box may be repeated by exiting and re-entering the box. The following example will illustrate the above:



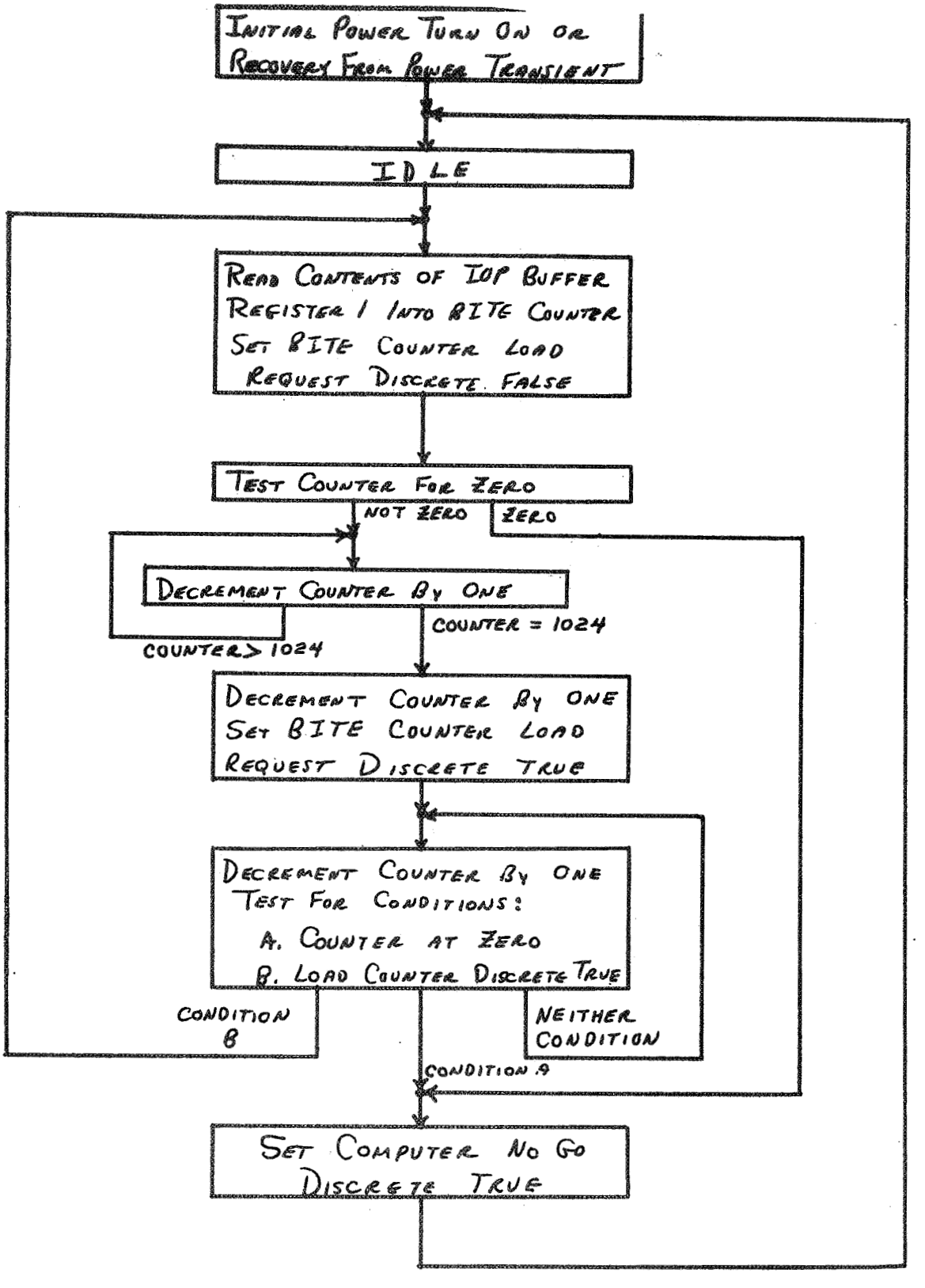
The circuit enters Box A, increments the counter and exits along one of the two paths determined by the state of the counter at the exit time. If the condition that the counter does not equal 5 exists, the circuit re-enters Box A. Once the condition of the counter equals five is met, the circuit enters Box B. The circuit will perform the actions of Box B but will not exit until the condition is met at the time that the clock signal allows the circuit to change state. Between boxes C and D, there is only one path and with no legend, this indicates that the circuit will exit Box C and enter Box D at the next time the clock signal permits.

4.2 Charts

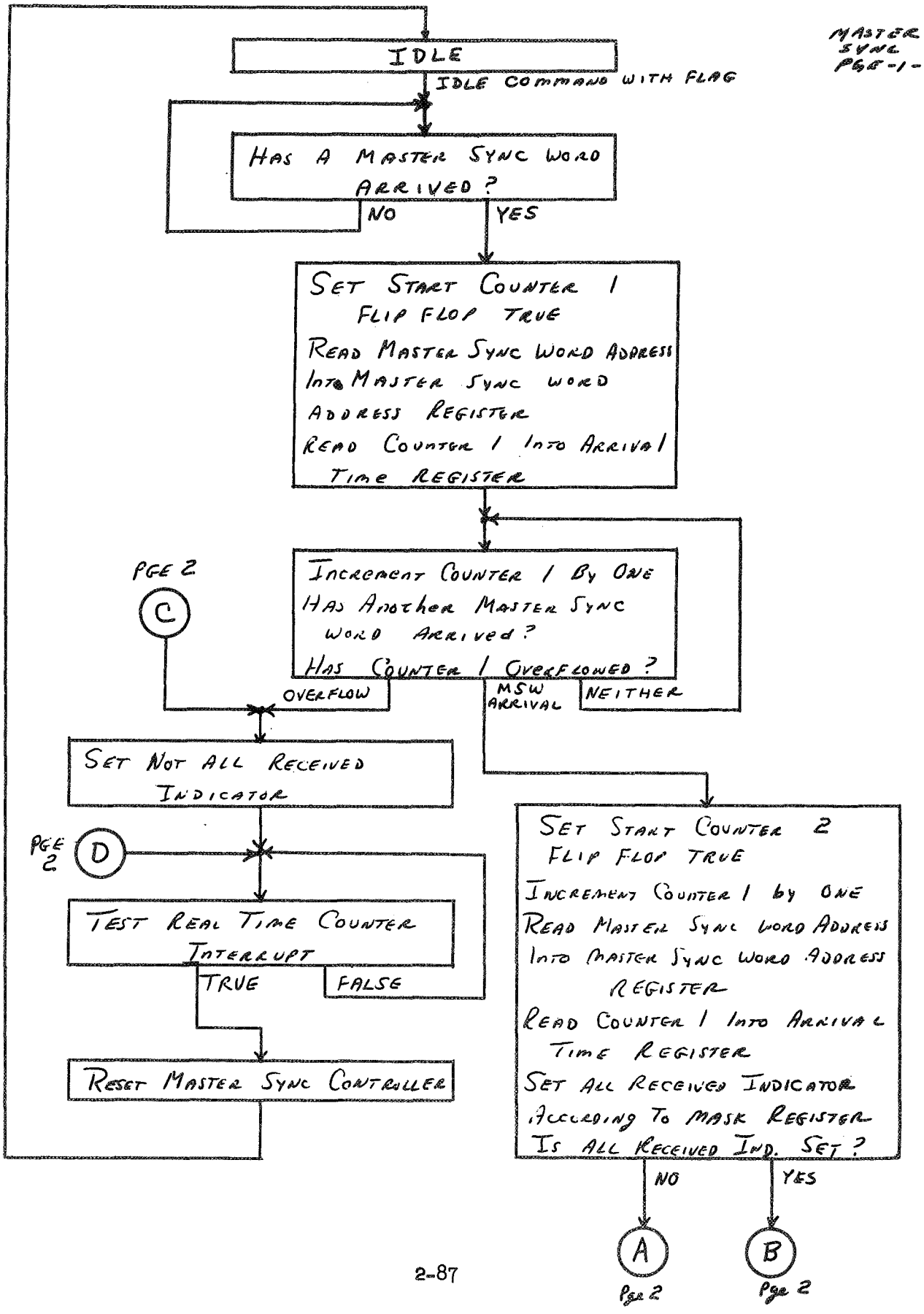
4.2.1 Real Time Counter

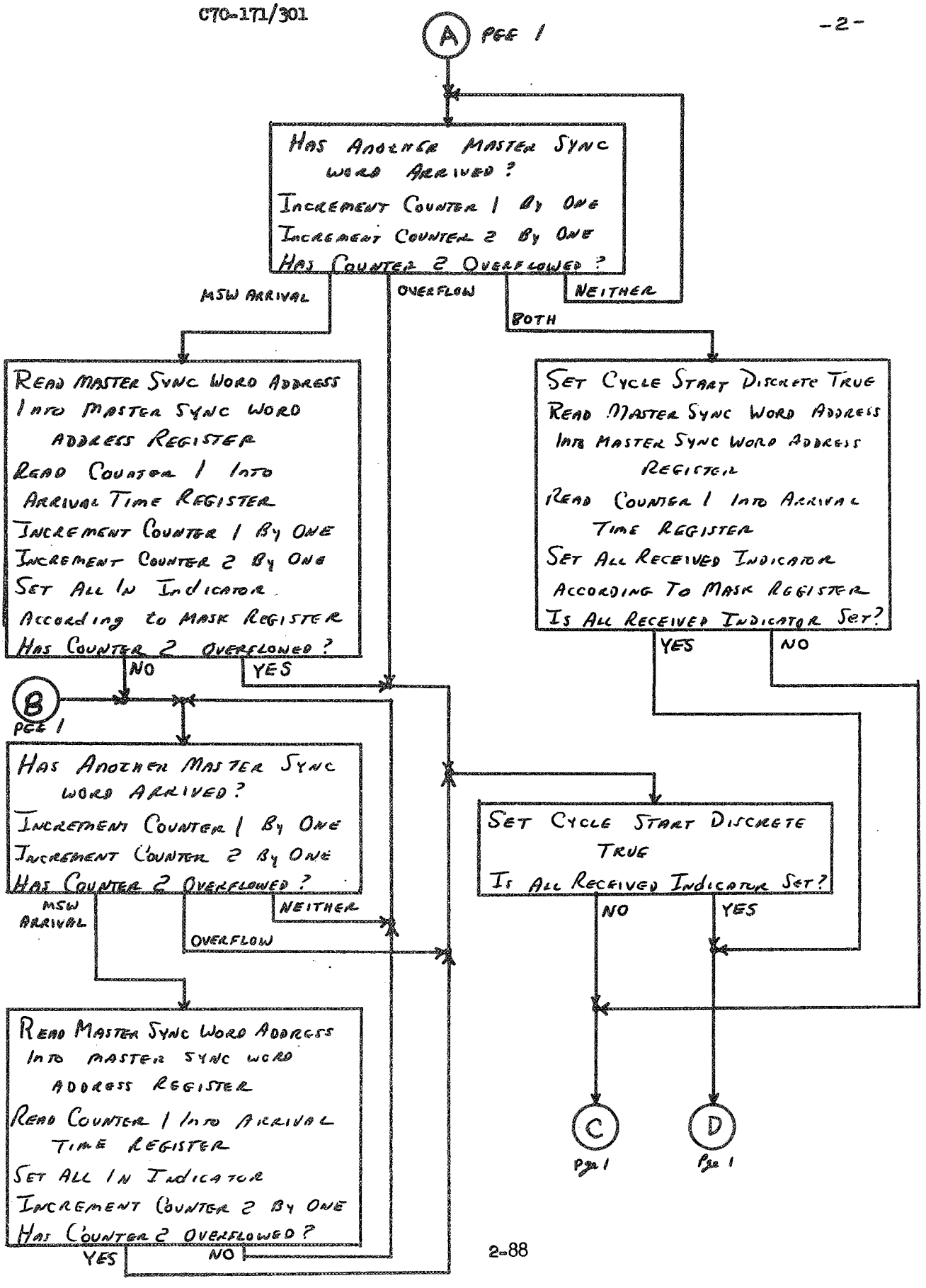


4.2.2 **BITE Counter**

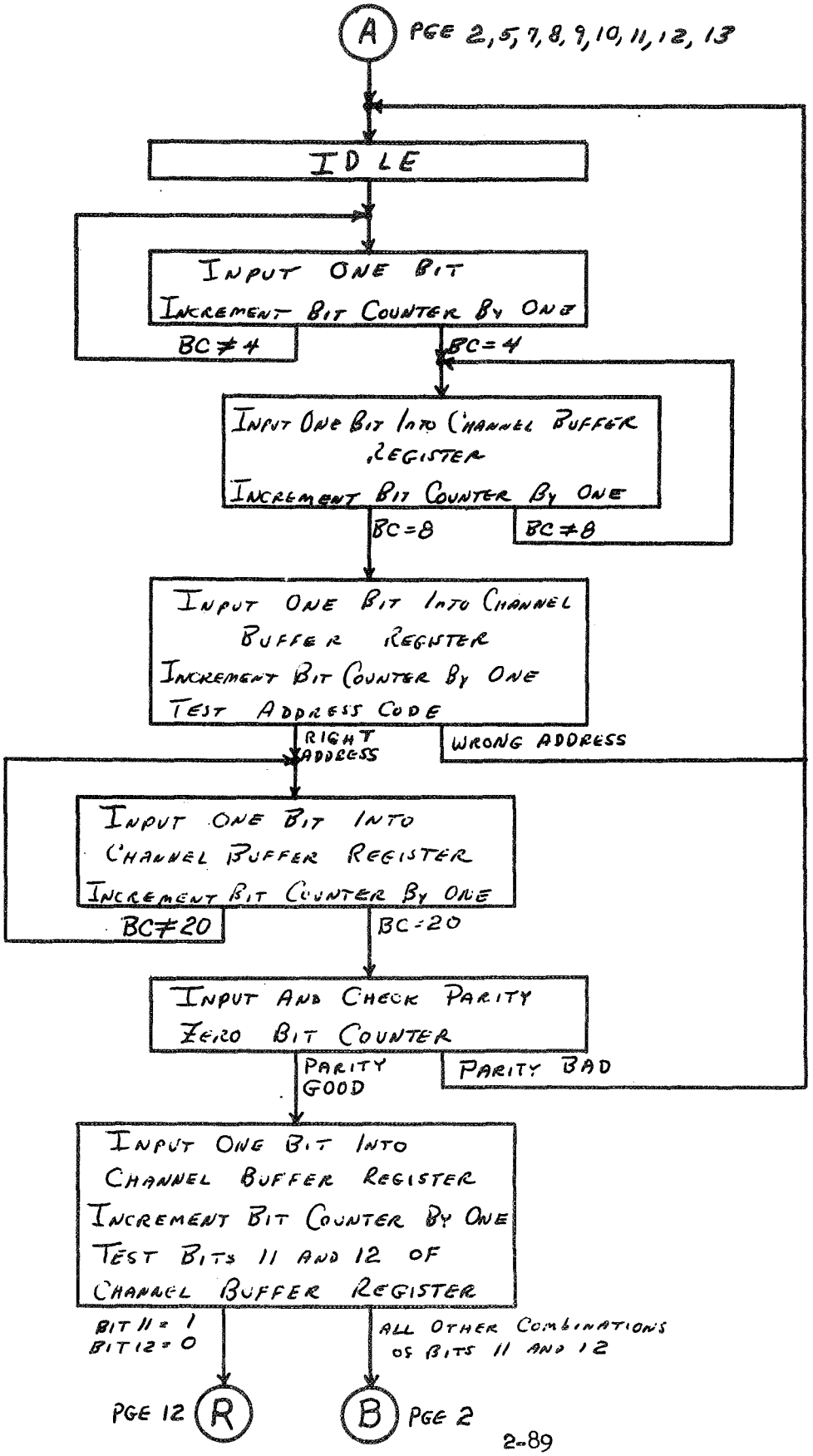


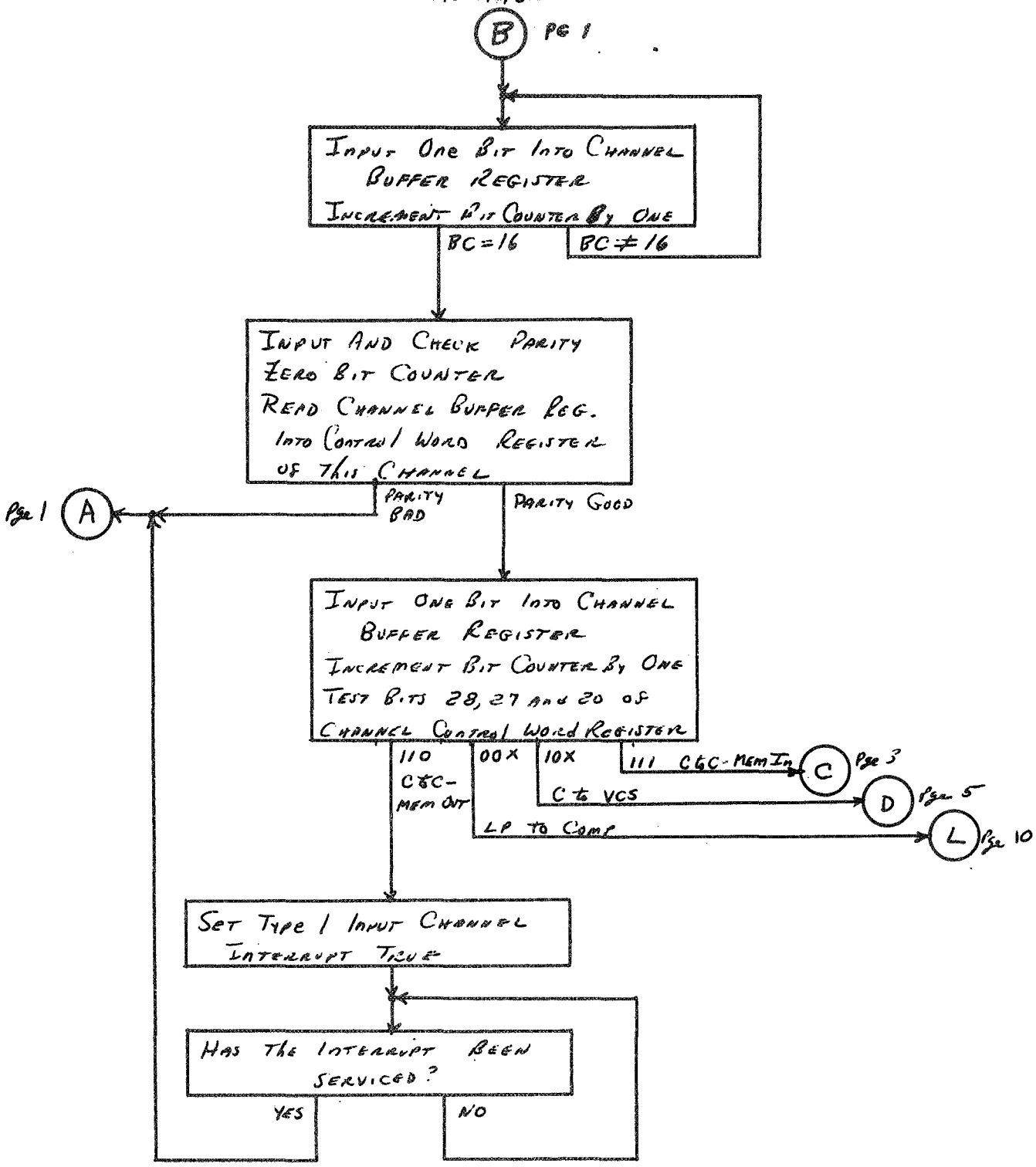
4.2.3 Master Sync Controller

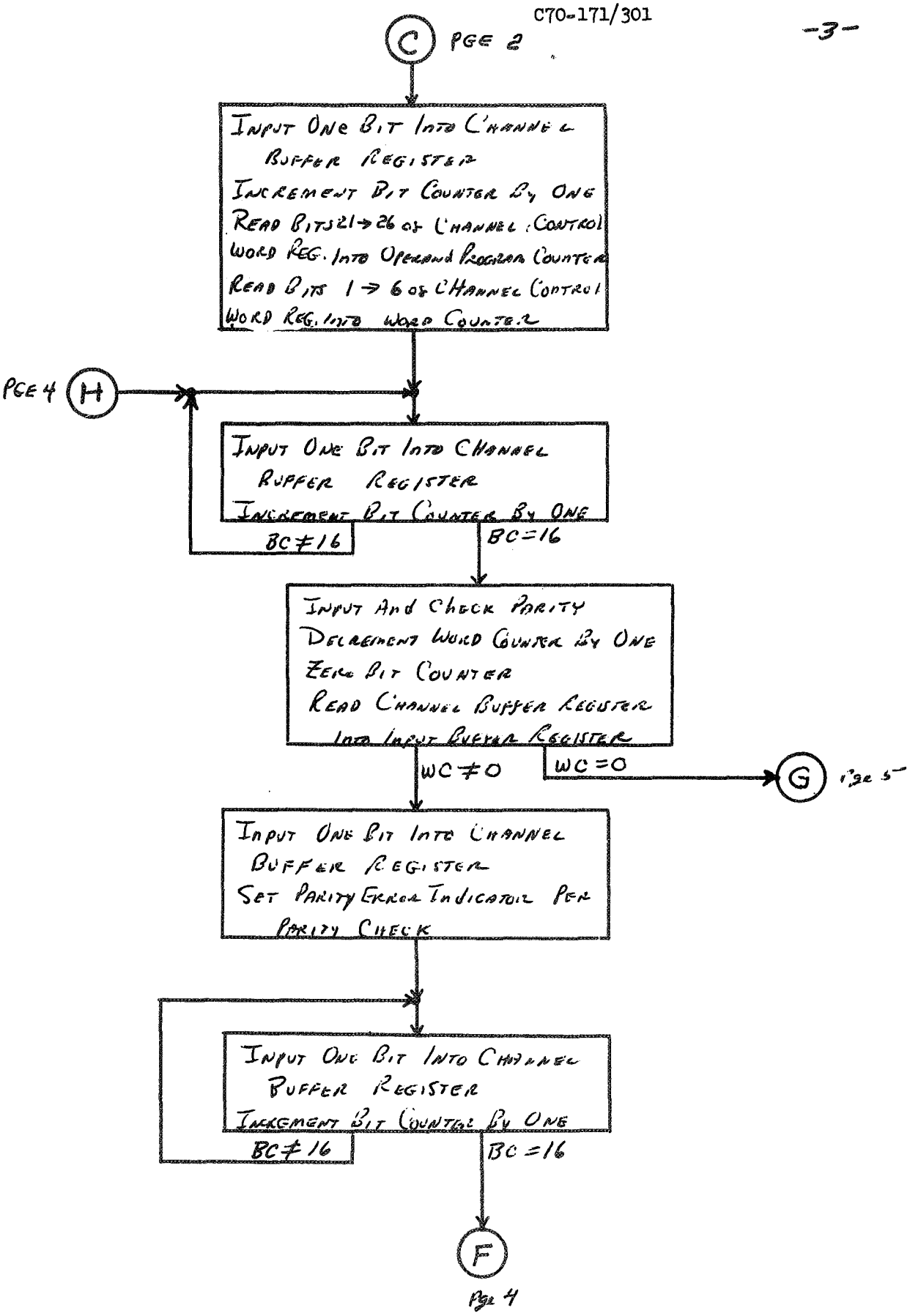


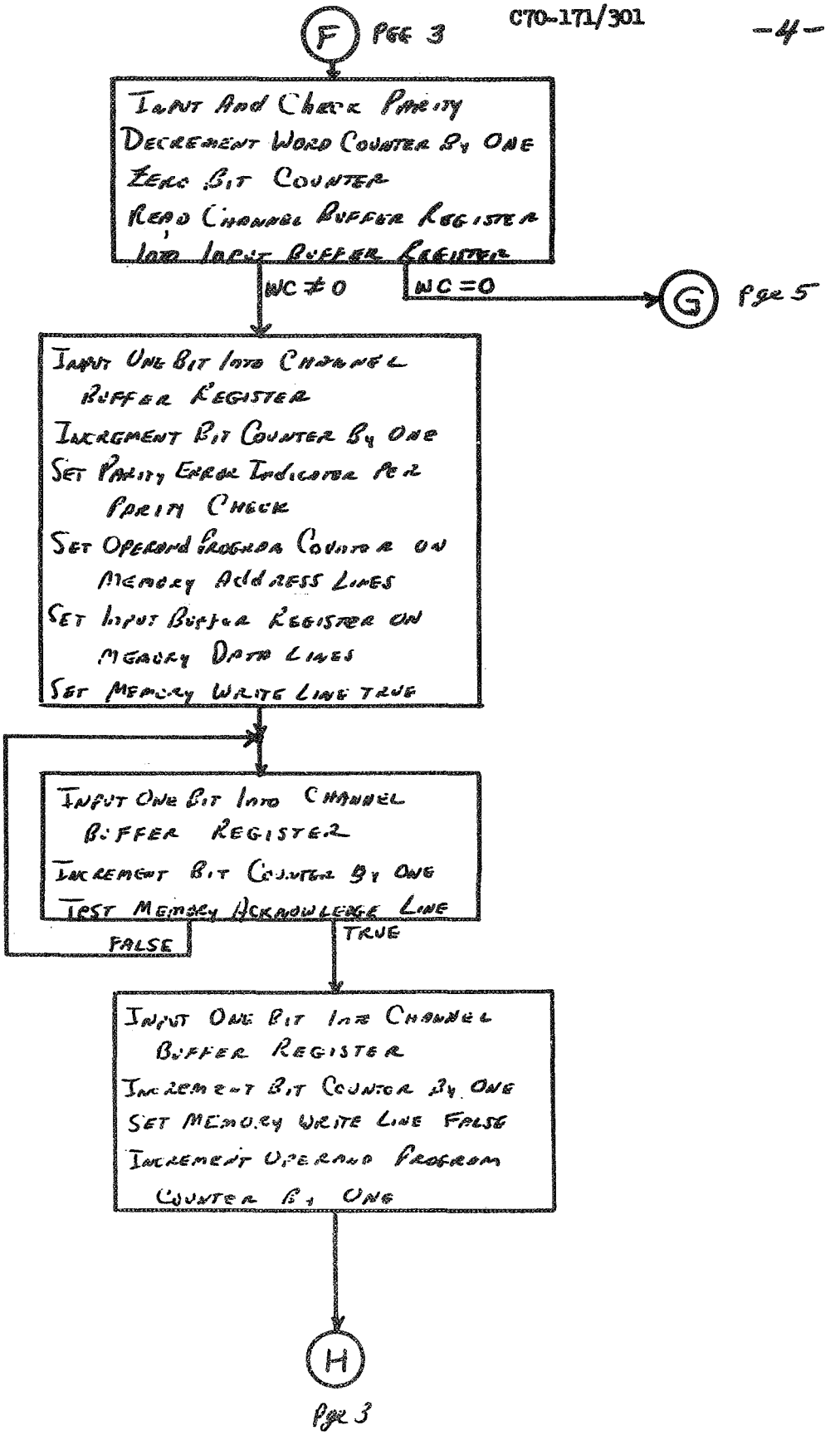

A
PGE 1
HAS ANOTHER MASTER SYNC WORD ARRIVED?
INCREMENT COUNTER 1 BY ONE
INCREMENT COUNTER 2 BY ONE
HAS COUNTER 2 OVERFLOWED?
MSW ARRIVAL
OVERFLOW
NEITHER
BOTH
READ MASTER SYNC WORD ADDRESS INTO MASTER SYNC WORD ADDRESS REGISTER
READ COUNTER 1 INTO ARRIVAL TIME REGISTER
INCREMENT COUNTER 1 BY ONE
INCREMENT COUNTER 2 BY ONE
SET ALL IN INDICATOR ACCORDING TO MASK REGISTER
HAS COUNTER 2 OVERFLOWED?
NO
YES
SET CYCLE START DISCRETE TRUE
READ MASTER SYNC WORD ADDRESS INTO MASTER SYNC WORD ADDRESS REGISTER
READ COUNTER 1 INTO ARRIVAL TIME REGISTER
SET ALL RECEIVED INDICATOR ACCORDING TO MASK REGISTER
IS ALL RECEIVED INDICATOR SET?
YES
NO
B
PGE 1
HAS ANOTHER MASTER SYNC WORD ARRIVED?
INCREMENT COUNTER 1 BY ONE
INCREMENT COUNTER 2 BY ONE
HAS COUNTER 2 OVERFLOWED?
MSW ARRIVAL
NEITHER
OVERFLOW
SET CYCLE START DISCRETE TRUE
IS ALL RECEIVED INDICATOR SET?
NO
YES
READ MASTER SYNC WORD ADDRESS INTO MASTER SYNC WORD ADDRESS REGISTER
READ COUNTER 1 INTO ARRIVAL TIME REGISTER
SET ALL IN INDICATOR
INCREMENT COUNTER 2 BY ONE
HAS COUNTER 2 OVERFLOWED?
YES
NO
C
PGE 1
D
PGE 1

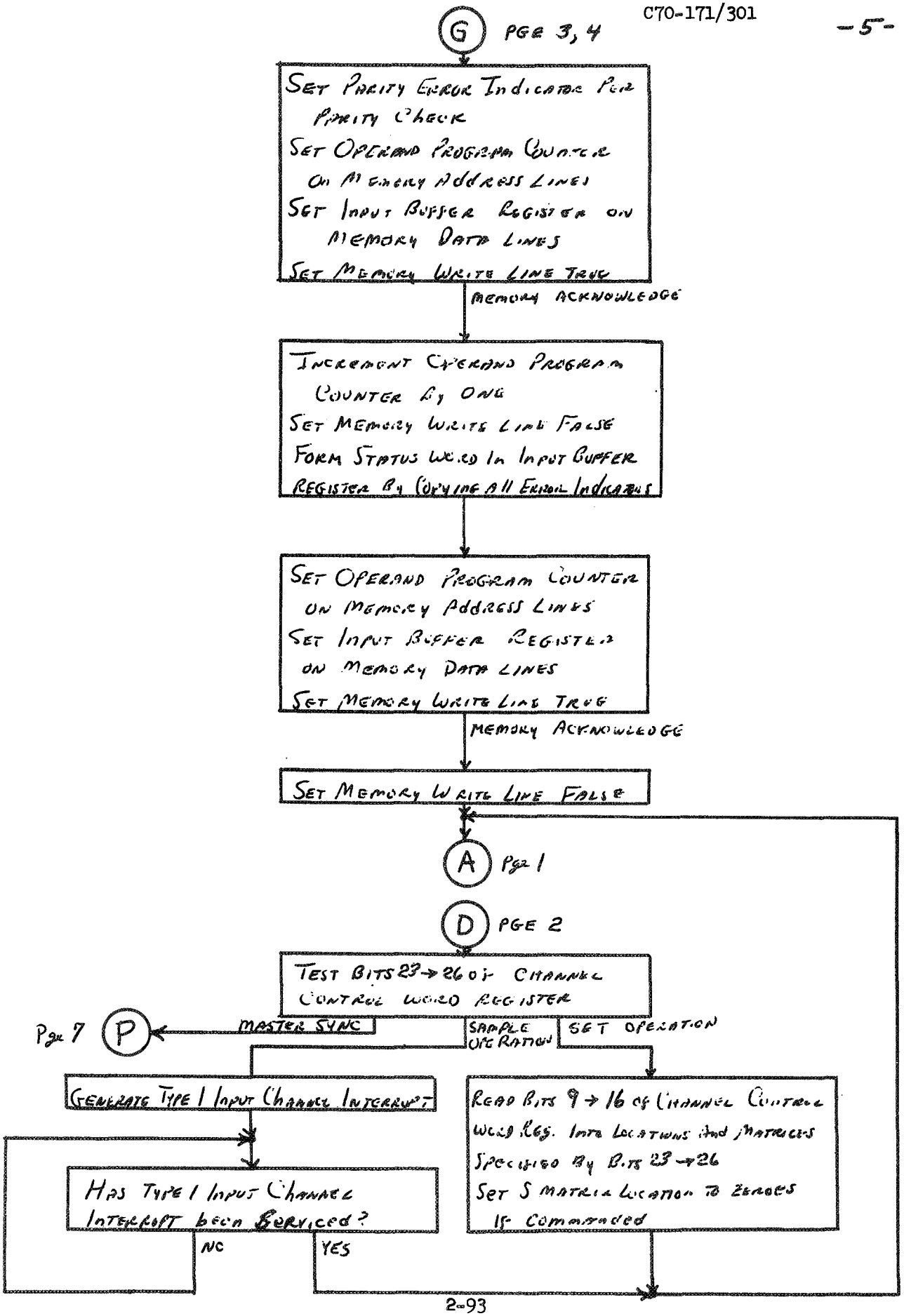
4.2.4 Type 1 Input Channel

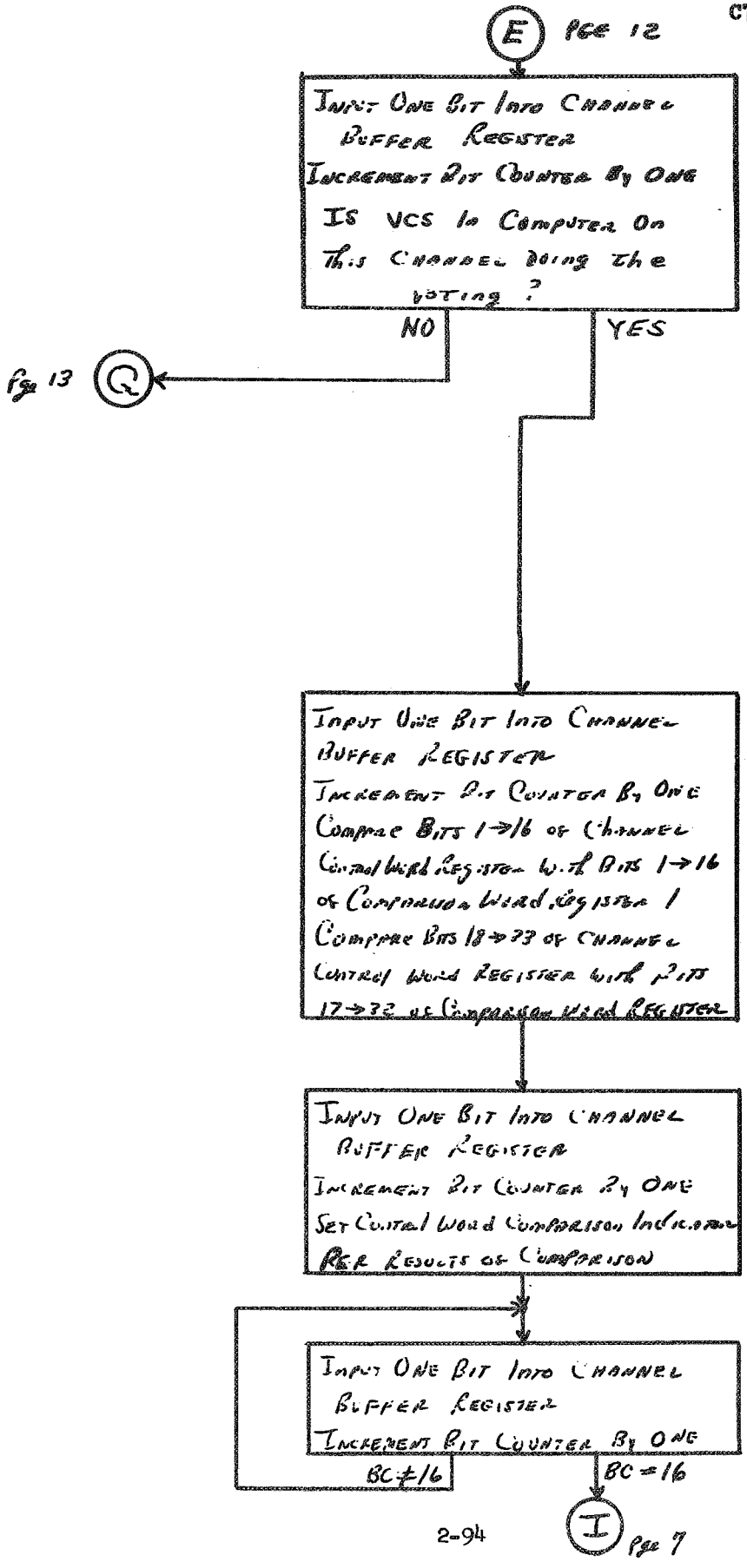


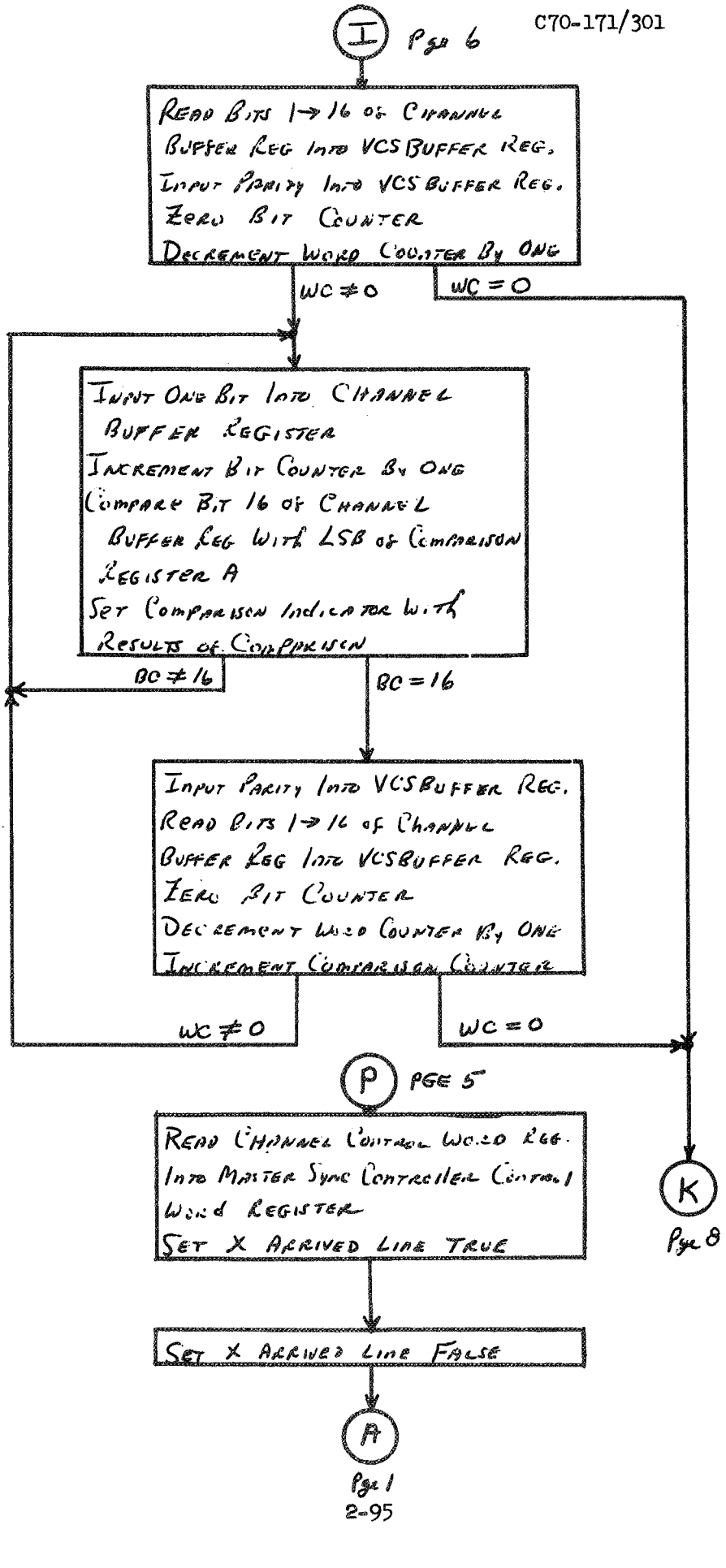

B
Pg 1
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
BC=16
BC≠16
Input And Check Parity
Zero Bit Counter
Read Channel Buffer Reg. Into Control Word Register of This Channel
Pge 1
A
Parity Bad
Parity Good
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Test Bits 28, 27 and 20 of Channel Control Word Register
110 C&C-Mem Out
00X
10X
111 C&C-Mem In
C
Pge 3
C to VCS
D
Pge 5
LP to Comp
L
Pge 10
Set Type 1 Input Channel Interrupt True
Has The Interrupt Been Serviced?
Yes
No

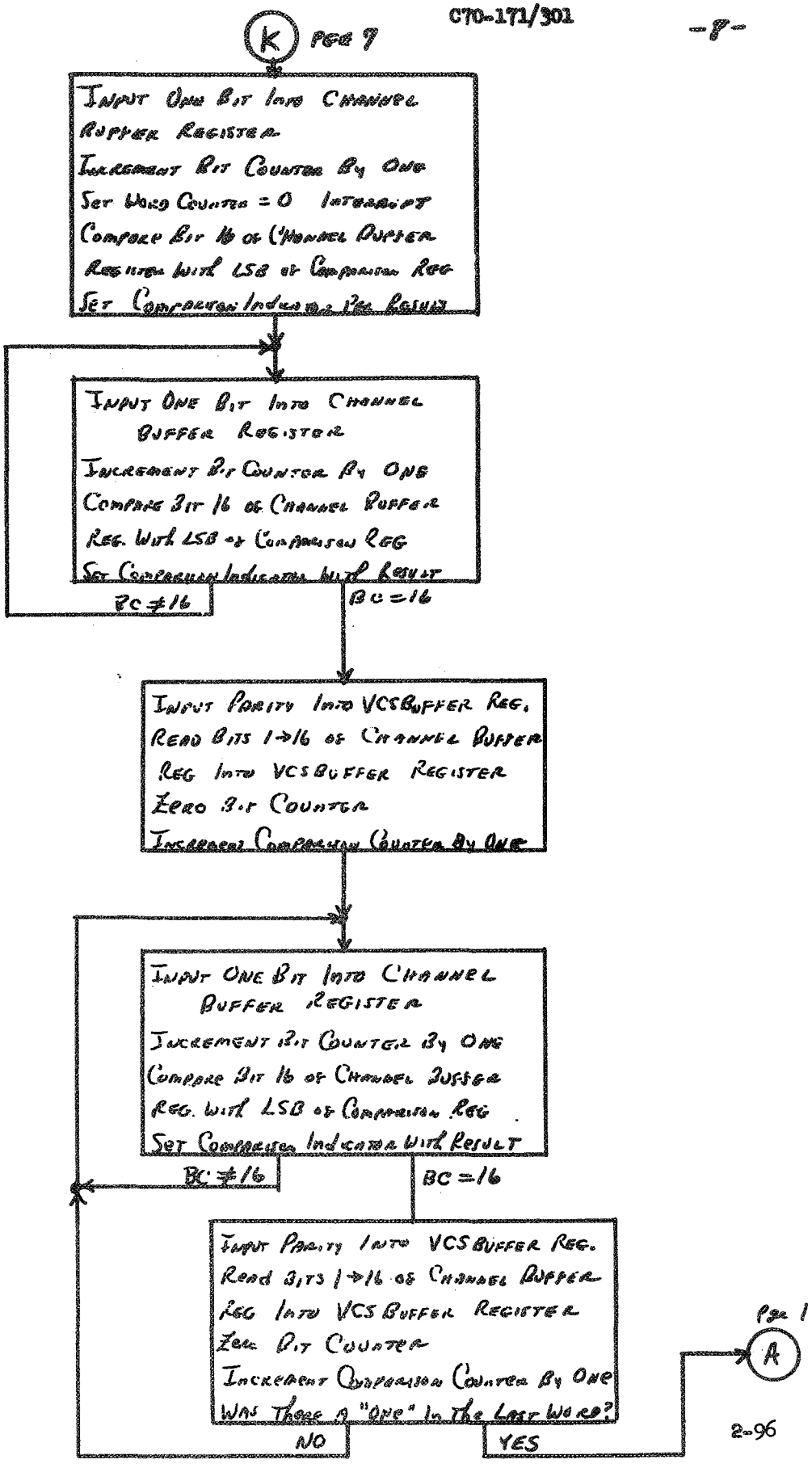

C
PGE 2
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
READ BITS 21 → 26 OF CHANNEL CONTROL WORD REG. INTO OPERAND PROGRAM COUNTER
READ BITS 1 → 6 OF CHANNEL CONTROL WORD REG. INTO WORD COUNTER
PGE 4
H
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
INPUT AND CHECK PARITY
DECREMENT WORD COUNTER BY ONE
ZERO BIT COUNTER
READ CHANNEL BUFFER REGISTER INTO INPUT BUFFER REGISTER
WC ≠ 0
WC = 0
G
PGE 5
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
SET PARITY ERROR INDICATOR PER PARITY CHECK
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
F
PGE 4


F
Pge 3
Input And Check Parity
Decrement Word Counter By One
Zero Bit Counter
Read Channel Buffer Register
Load Input Buffer Register
WC ≠ 0
WC = 0
G
Pge 5
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Set Parity Error Indicator For Parity Check
Set Operand Program Counter On Memory Address Lines
Set Input Buffer Register On Memory Data Lines
Set Memory Write Line True
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Test Memory Acknowledge Line
FALSE
TRUE
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Set Memory Write Line False
Increment Operand Program Counter By One
H
Pge 3

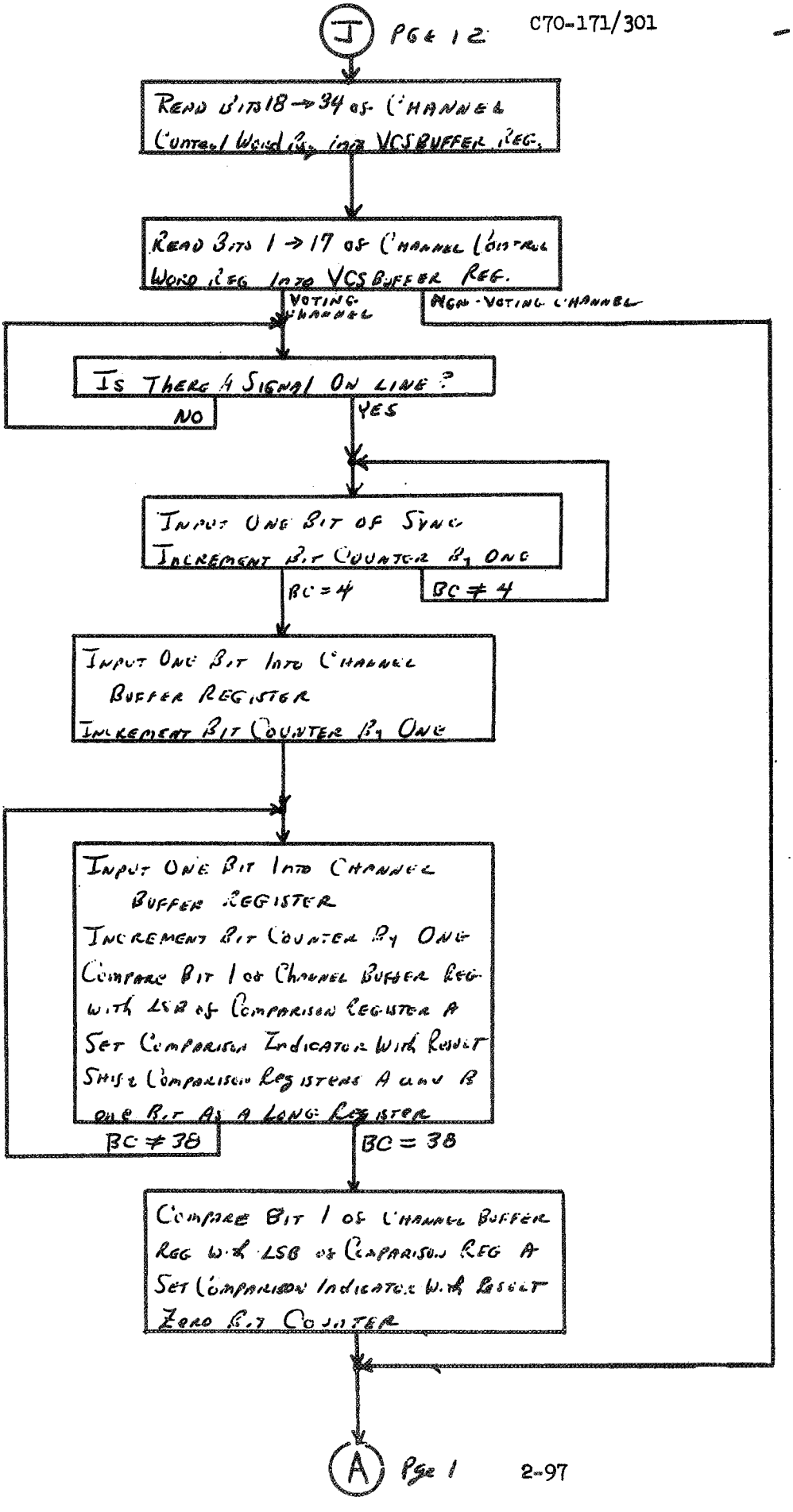

G PGE 3, 4
SET PARITY ERROR INDICATOR FOR PARITY CHECK
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET INPUT BUFFER REGISTER ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SET MEMORY WRITE LINE FALSE
FORM STATUS WORD IN INPUT BUFFER REGISTER BY ORING ALL ERROR INDICATORS
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET INPUT BUFFER REGISTER ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
SET MEMORY WRITE LINE FALSE
A Pge 1
D PGE 2
TEST BITS 23 → 26 OF CHANNEL CONTROL WORD REGISTER
Pge 7 P
MASTER SYNC
SAMPLE OPERATION
SET OPERATION
GENERATE TYPE I INPUT CHANNEL INTERRUPT
HAS TYPE I INPUT CHANNEL INTERRUPT been Serviced?
NO
YES
READ BITS 9 → 16 OF CHANNEL CONTROL WORD REG. INTO LOCATIONS and MATRICES SPECIFIED BY BITS 23 → 26
SET S MATRIX LOCATION TO ZEROES IF COMMANDED

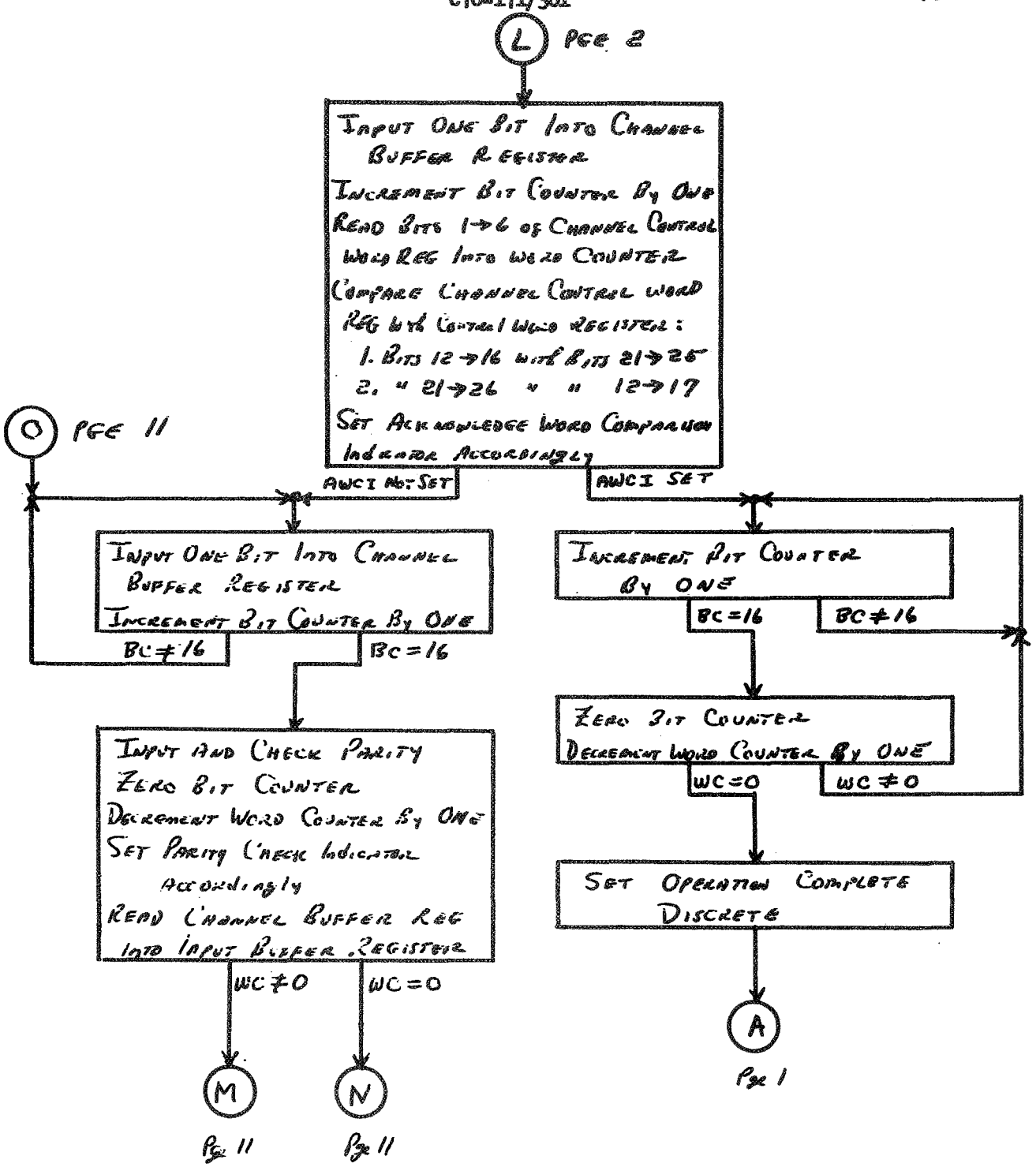

E
Pge 12
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Is VCS In Computer On This Channel Doing The Voting ?
NO
YES
Pge 13
Q
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Compare Bits 1→16 of Channel Control Word Register W.th Bits 1→16 of Comparison Word Register /
Compare Bits 18→33 of Channel Control Word Register With Bits 17→32 of Comparison Word Register
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Set Control Word Comparison Indicators Per Results of Comparison
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
BC ≠ 16
BC = 16
I
Pge 7

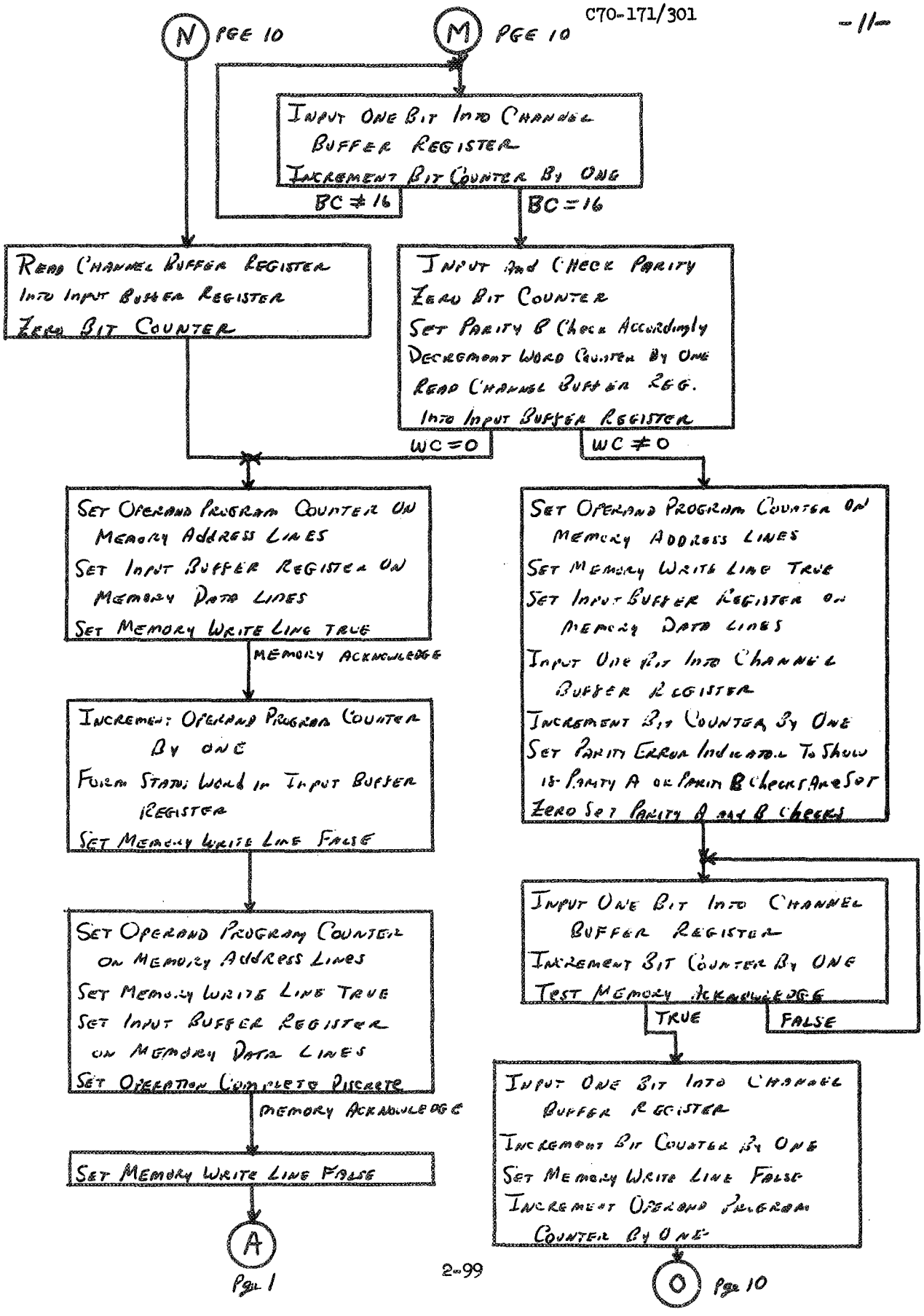

I
Pge 6
READ BITS 1→16 OF CHANNEL BUFFER REG INTO VCS BUFFER REG.
INPUT PARITY INTO VCS BUFFER REG.
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
COMPARE BIT 16 OF CHANNEL BUFFER REG WITH LSB OF COMPARISON REGISTER A
SET COMPARISON INDICATOR WITH RESULTS OF COMPARISON
BC ≠ 16
BC = 16
INPUT PARITY INTO VCS BUFFER REG.
READ BITS 1→16 OF CHANNEL BUFFER REG INTO VCS BUFFER REG.
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
INCREMENT COMPARISON COUNTER
WC ≠ 0
WC = 0
P
PGE 5
READ CHANNEL CONTROL WORD REG. INTO MASTER SYNC CONTROLLER CONTROL WORD REGISTER
SET X ARRIVED LINE TRUE
K
Pge 8
SET X ARRIVED LINE FALSE
A
Pge 1

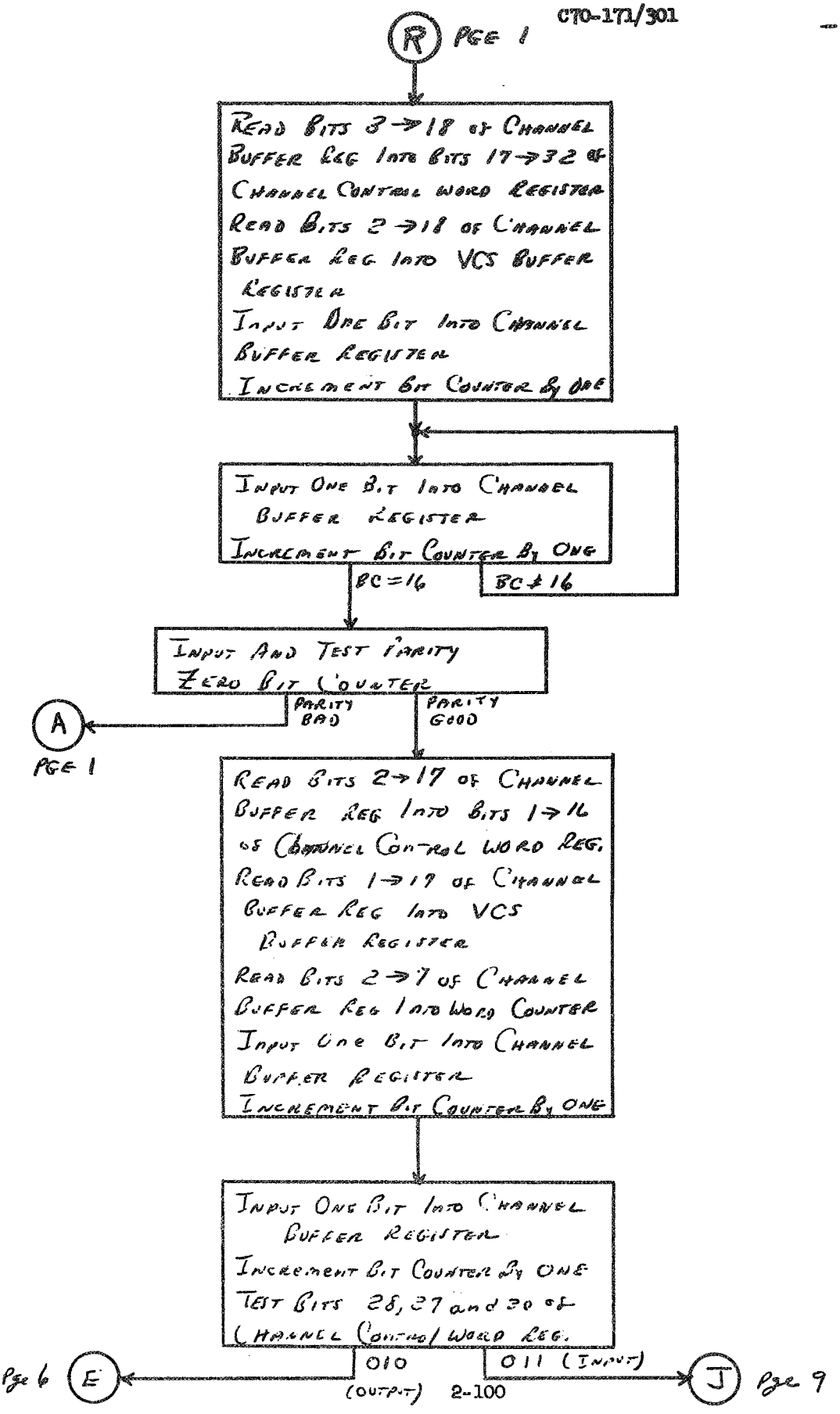

K
Pge 7
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Set Word Counter = 0 Interrupt
Compare Bit 16 of Channel Buffer Register with LSB of Comparison Reg
Set Comparison Indicators Per Result
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Compare Bit 16 of Channel Buffer Reg. With LSB of Comparison Reg
Set Comparison Indicators With Result
BC ≠ 16
BC = 16
Input Parity Into VCS Buffer Reg.
Read Bits 1 → 16 of Channel Buffer Reg Into VCS Buffer Register
Zero Bit Counter
Increment Comparison Counter By One
Input One Bit Into Channel Buffer Register
Increment Bit Counter By One
Compare Bit 16 of Channel Buffer Reg. With LSB of Comparison Reg
Set Comparison Indicators With Result
BC ≠ 16
BC = 16
Input Parity Into VCS Buffer Reg.
Read Bits 1 → 16 of Channel Buffer Reg Into VCS Buffer Register
Zero Bit Counter
Increment Comparison Counter By One
Was There a "One" In the Last Word?
NO
YES
Pge 1
A

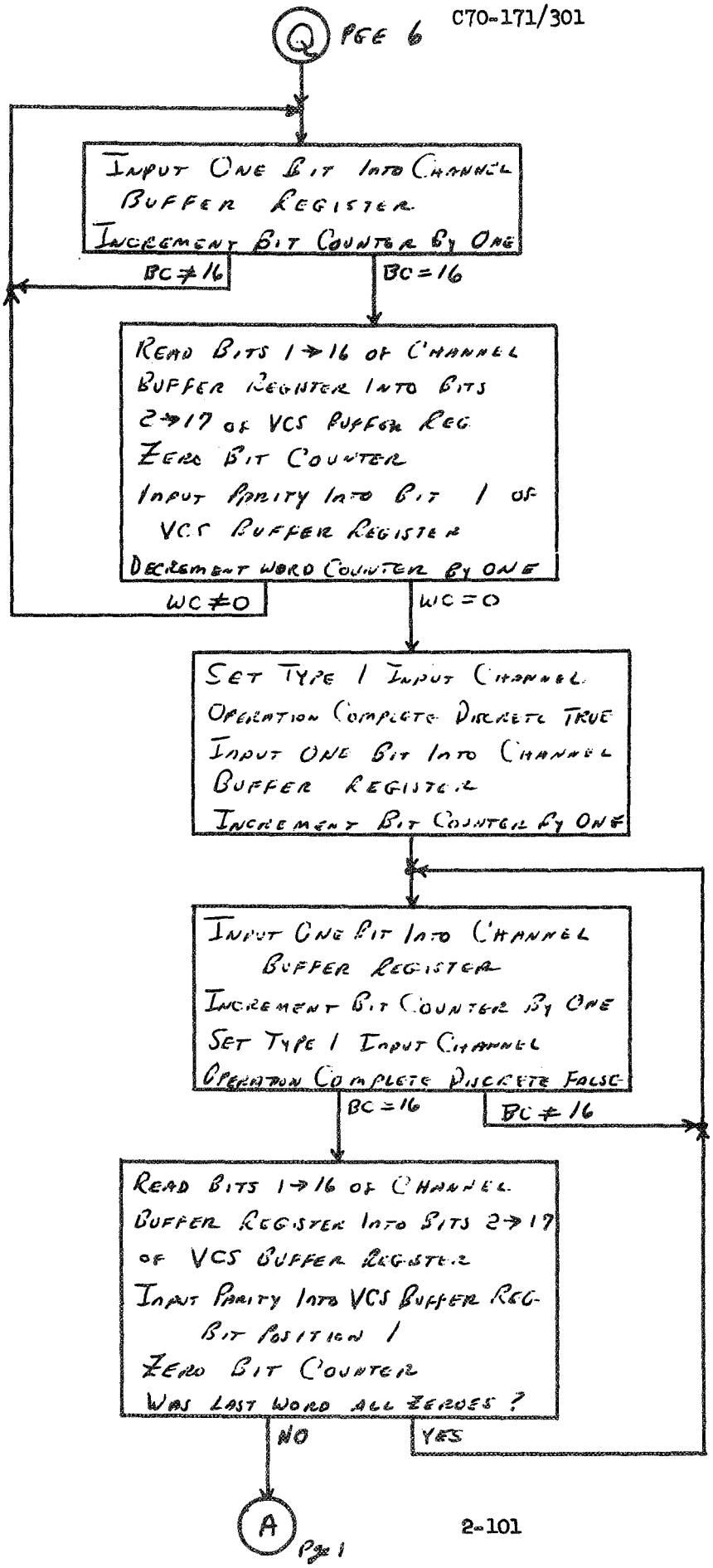
2-96


J
PGE 12
READ BITS 18 → 34 OF CHANNEL CONTROL WORD REG. INTO VCS BUFFER REG.
READ BITS 1 → 17 OF CHANNEL CONTROL WORD REG INTO VCS BUFFER REG.
VOTING CHANNEL
NON-VOTING CHANNEL
IS THERE A SIGNAL ON LINE ?
NO
YES
INPUT ONE BIT OF SYNC
INCREMENT BIT COUNTER BY ONE
BC = 4
BC ≠ 4
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
COMPARE BIT 1 OF CHANNEL BUFFER REG. WITH LSB OF COMPARISON REGISTER A
SET COMPARISON INDICATOR WITH RESULT
SHIFT COMPARISON REGISTERS A AND B ONE BIT AS A LONG REGISTER
BC ≠ 38
BC = 38
COMPARE BIT 1 OF CHANNEL BUFFER REG W.R. LSB OF COMPARISON REG A
SET COMPARISON INDICATOR W.R. RESULT
ZERO BIT COUNTER
A
PGE 1

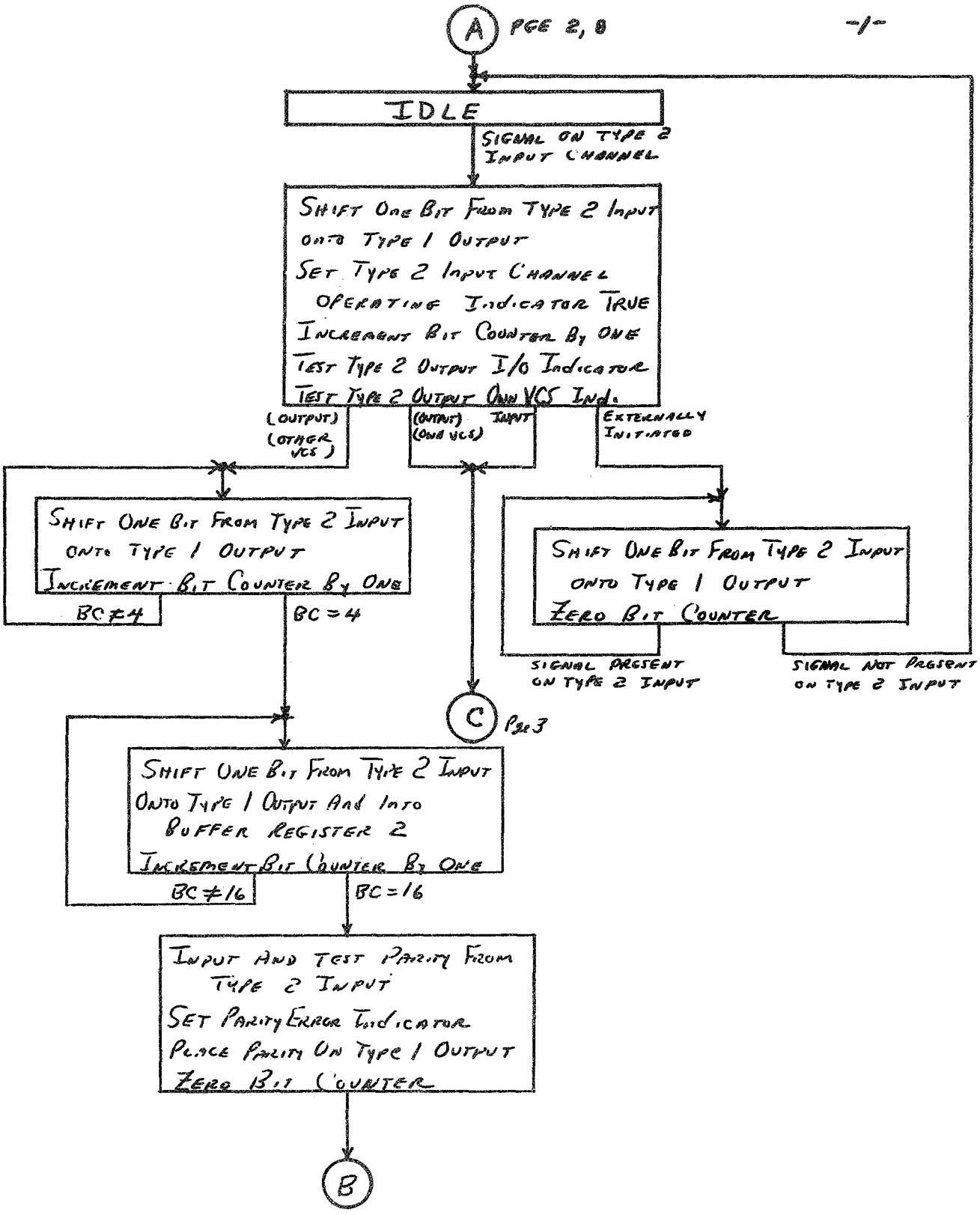

L
PGE 2
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
READ BITS 1→6 OF CHANNEL CONTROL WORD REG INTO WORD COUNTER
COMPARE CHANNEL CONTROL WORD REG WITH CONTROL WORD REGISTER:
1. BITS 12→16 WITH BITS 21→25
2. " 21→26 " " 12→17
SET ACKNOWLEDGE WORD COMPARISON INDICATOR ACCORDINGLY
AWCI NOT SET
AWCI SET
O
PGE 11
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
BC≠16
BC=16
INCREMENT BIT COUNTER BY ONE
BC=16
BC≠16
INPUT AND CHECK PARITY
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
SET PARITY CHECK INDICATOR ACCORDINGLY
READ CHANNEL BUFFER REG INTO INPUT BUFFER REGISTER
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
WC=0
WC≠0
WC≠0
WC=0
SET OPERATION COMPLETE DISCRETE
M
PGE 11
N
PGE 11
A
PGE 1

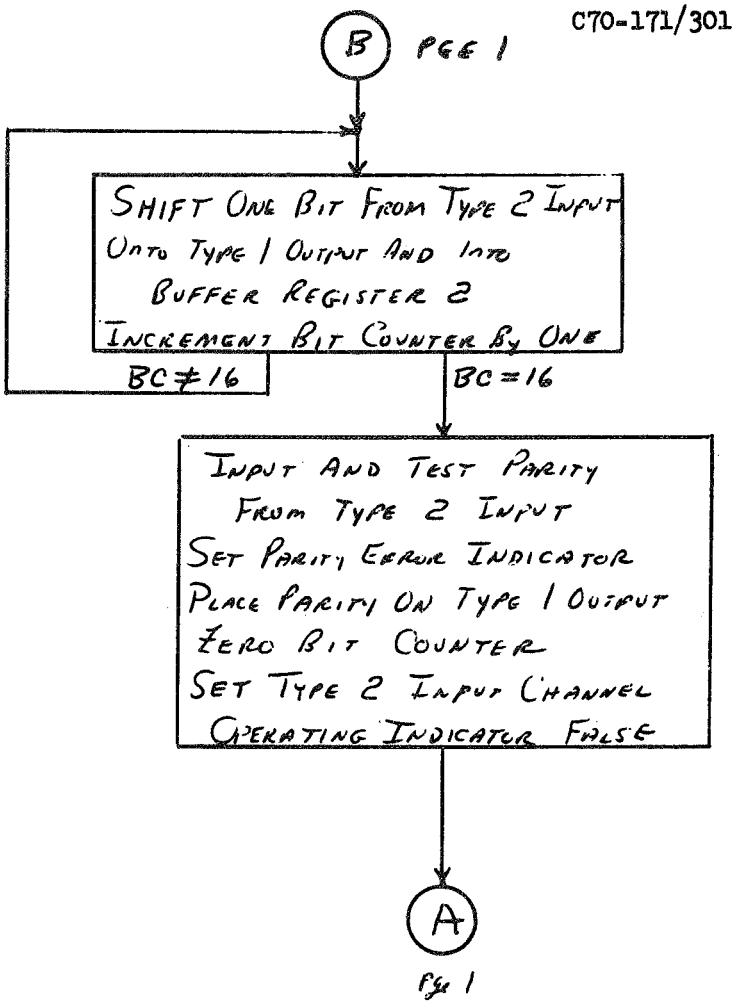

N PGE 10
M PGE 10
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
READ CHANNEL BUFFER REGISTER INTO INPUT BUFFER REGISTER
ZERO BIT COUNTER
INPUT AND CHECK PARITY
ZERO BIT COUNTER
SET PARITY B CHECK ACCORDINGLY
DECREMENT WORD COUNTER BY ONE
READ CHANNEL BUFFER REG. INTO INPUT BUFFER REGISTER
WC = 0
WC ≠ 0
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET INPUT BUFFER REGISTER ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
FORM STATUS WORD IN INPUT BUFFER REGISTER
SET MEMORY WRITE LINE FALSE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY WRITE LINE TRUE
SET INPUT BUFFER REGISTER ON MEMORY DATA LINES
SET OPERATION COMPLETE DISCRETE
MEMORY ACKNOWLEDGE
SET MEMORY WRITE LINE FALSE
A
Pge 1
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY WRITE LINE TRUE
SET INPUT BUFFER REGISTER ON MEMORY DATA LINES
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
SET PARITY ERROR INDICATOR TO SHOW IF PARITY A OR PARITY B CHECKS ARE SET
ZERO SET PARITY A AND B CHECKS
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
TRUE
FALSE
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
SET MEMORY WRITE LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
O
Pge 10

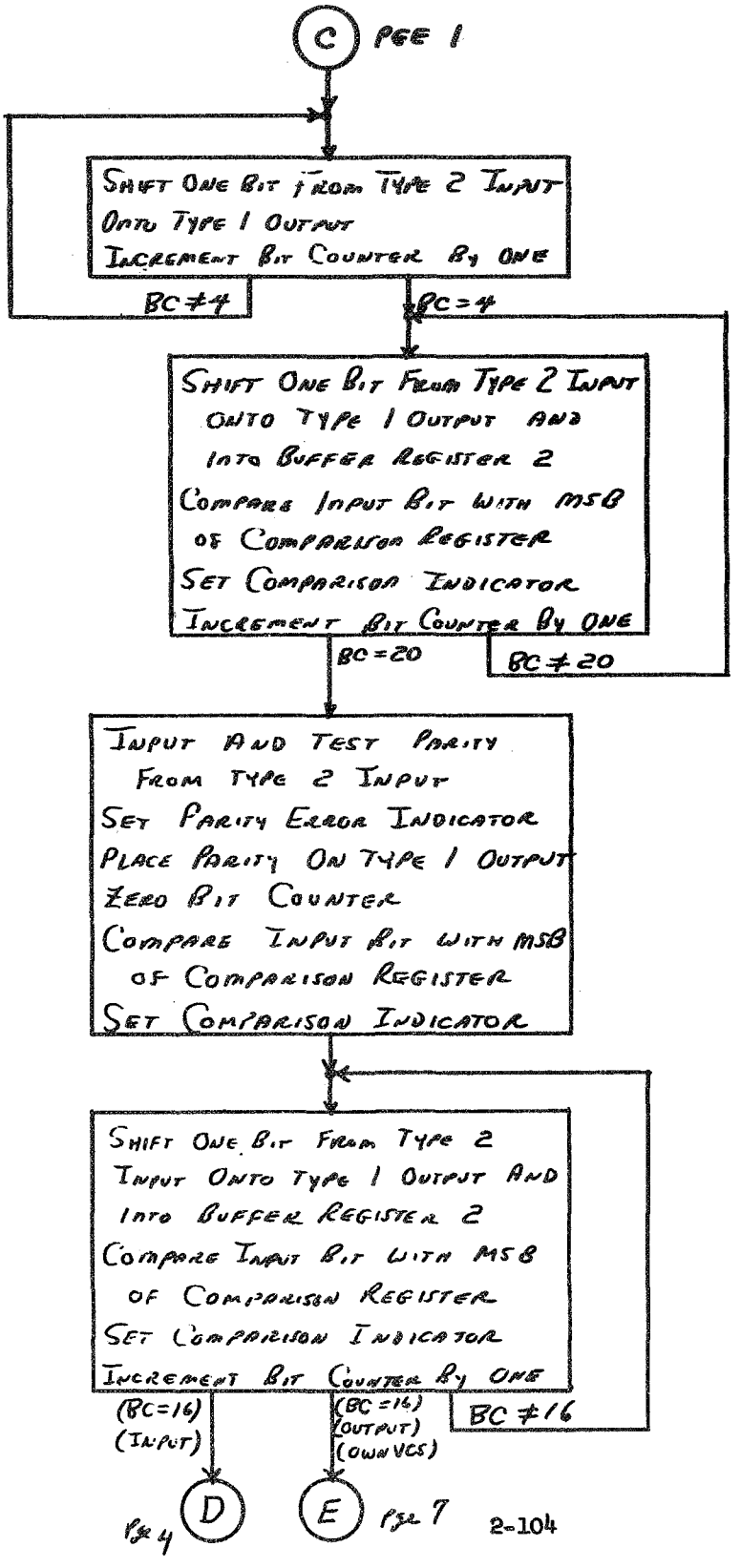

R
PGE 1
READ BITS 3 → 18 of CHANNEL BUFFER REG INTO BITS 17 → 32 of CHANNEL CONTROL WORD REGISTER
READ BITS 2 → 18 of CHANNEL BUFFER REG INTO VCS BUFFER REGISTER
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
INPUT AND TEST PARITY
ZERO BIT COUNTER
PARITY BAD
PARITY GOOD
A
PGE 1
READ BITS 2 → 17 of CHANNEL BUFFER REG INTO BITS 1 → 16 of CHANNEL CONTROL WORD REG.
READ BITS 1 → 17 of CHANNEL BUFFER REG INTO VCS BUFFER REGISTER
READ BITS 2 → 7 of CHANNEL BUFFER REG INTO WORD COUNTER
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
TEST BITS 28, 27 and 30 of CHANNEL CONTROL WORD REG.
010 (OUTPUT)
011 (INPUT)
2-100
E
Pge 6
J
Pge 9

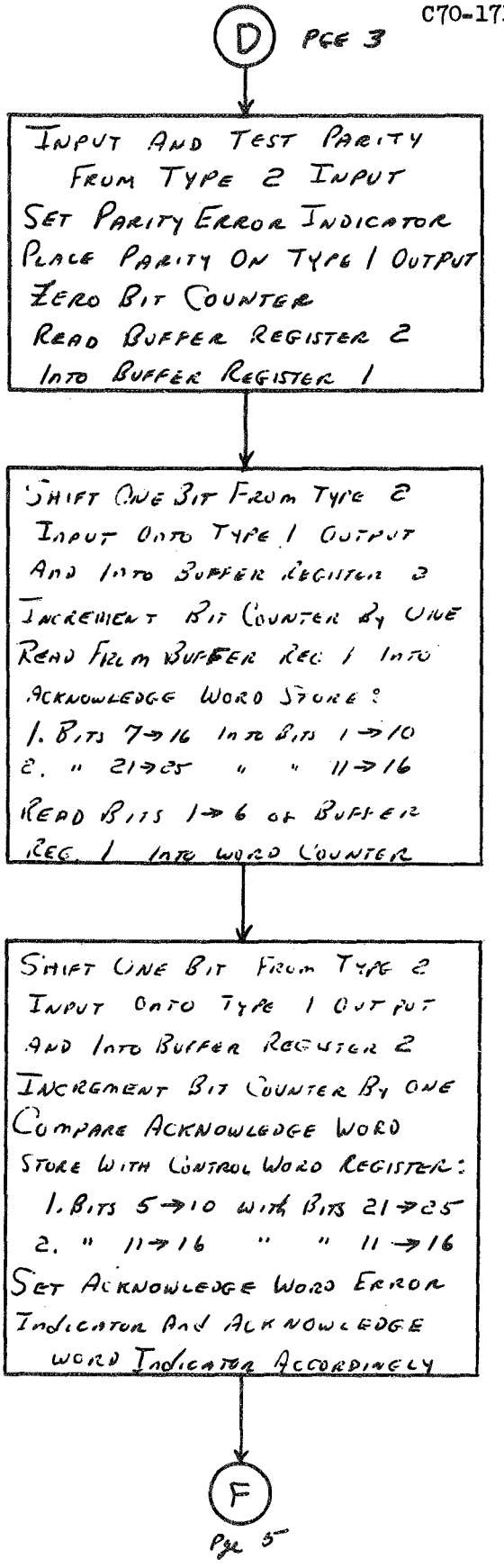

Q PGE 6
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREEMENT BIT COUNTER BY ONE
BC≠16
BC=16
READ BITS 1→16 OF CHANNEL BUFFER REGISTER INTO BITS 2→17 OF VCS BUFFER REG
ZERO BIT COUNTER
INPUT PARITY INTO BIT 1 OF VCS BUFFER REGISTER
DECREMENT WORD COUNTER BY ONE
WC≠0
WC=0
SET TYPE 1 INPUT CHANNEL OPERATION COMPLETE DISCRETE TRUE
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
INPUT ONE BIT INTO CHANNEL BUFFER REGISTER
INCREMENT BIT COUNTER BY ONE
SET TYPE 1 INPUT CHANNEL OPERATION COMPLETE DISCRETE FALSE
BC=16
BC≠16
READ BITS 1→16 OF CHANNEL BUFFER REGISTER INTO BITS 2→17 OF VCS BUFFER REGISTER
INPUT PARITY INTO VCS BUFFER REG BIT POSITION 1
ZERO BIT COUNTER
WAS LAST WORD ALL ZEROES ?
NO
YES
A PG 1

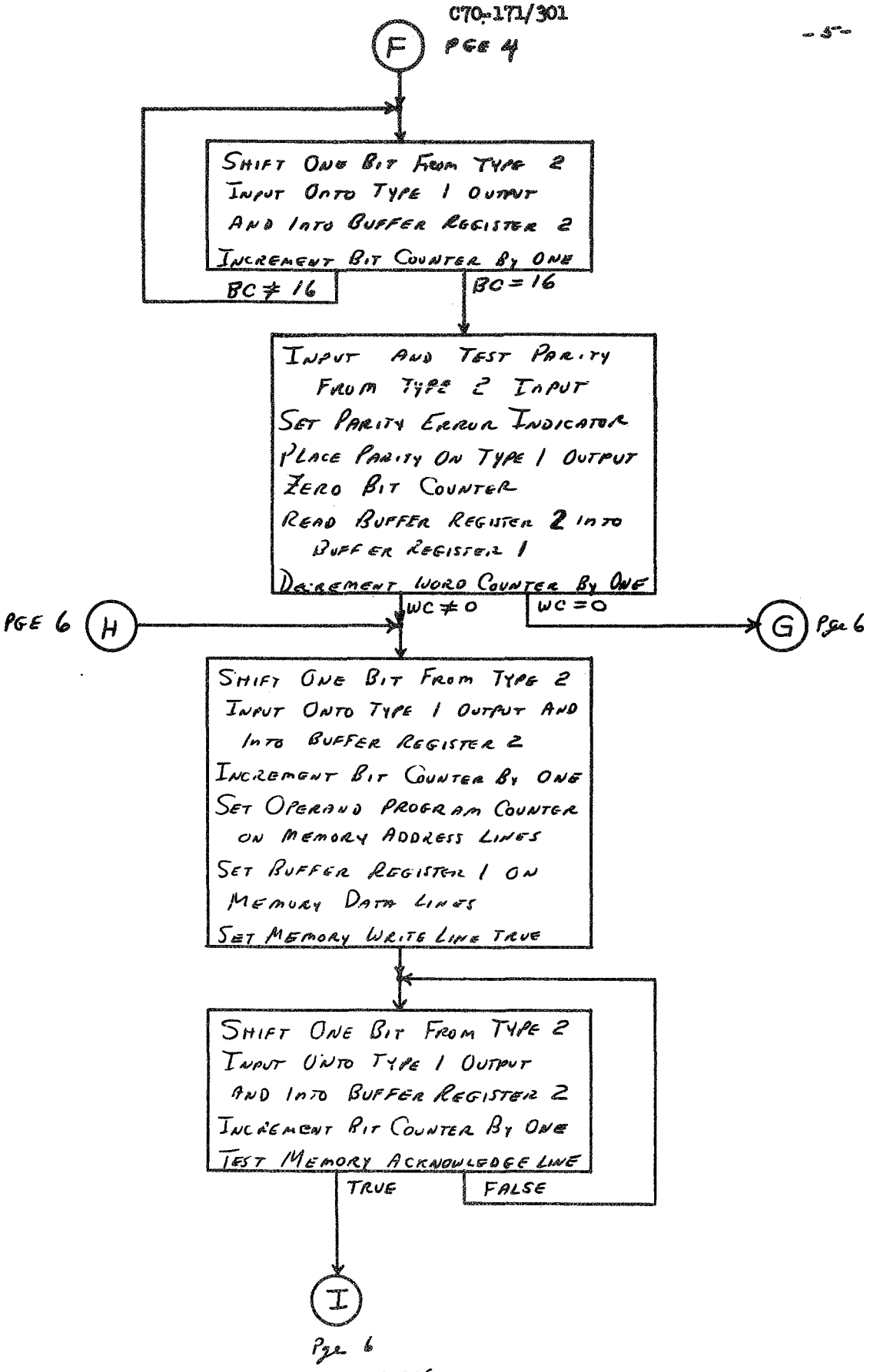
4.2.5 Type 2 Input Channel

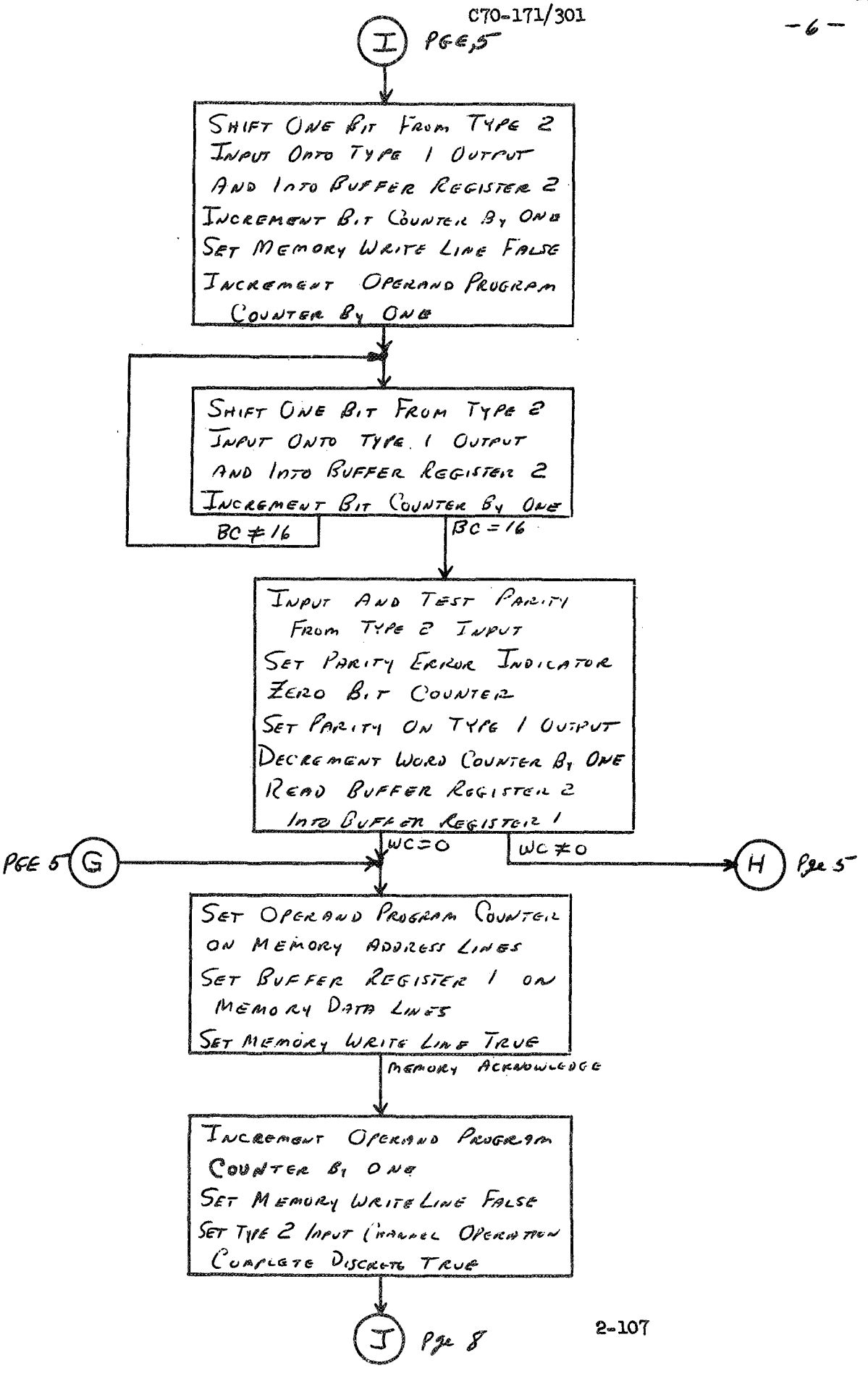


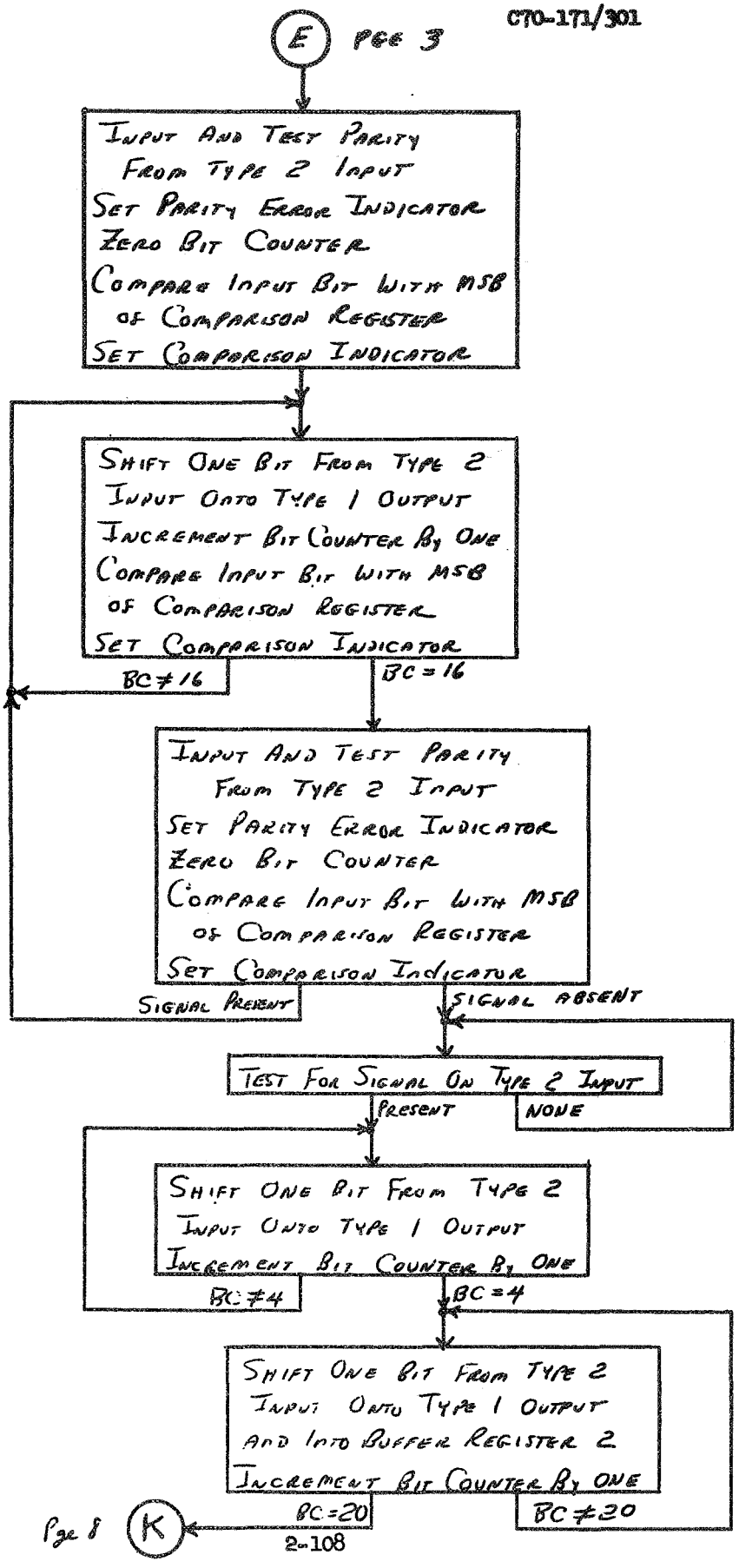

B
PGE 1
SHIFT ONE BIT FROM TYPE 2 INPUT
ONTO TYPE 1 OUTPUT AND INTO
BUFFER REGISTER 2
INCREMENT BIT COUNTER BY ONE
BC≠16
BC=16
INPUT AND TEST PARITY
FROM TYPE 2 INPUT
SET PARITY ERROR INDICATOR
PLACE PARITY ON TYPE 1 OUTPUT
ZERO BIT COUNTER
SET TYPE 2 INPUT CHANNEL
OPERATING INDICATOR FALSE
A
Pge 1

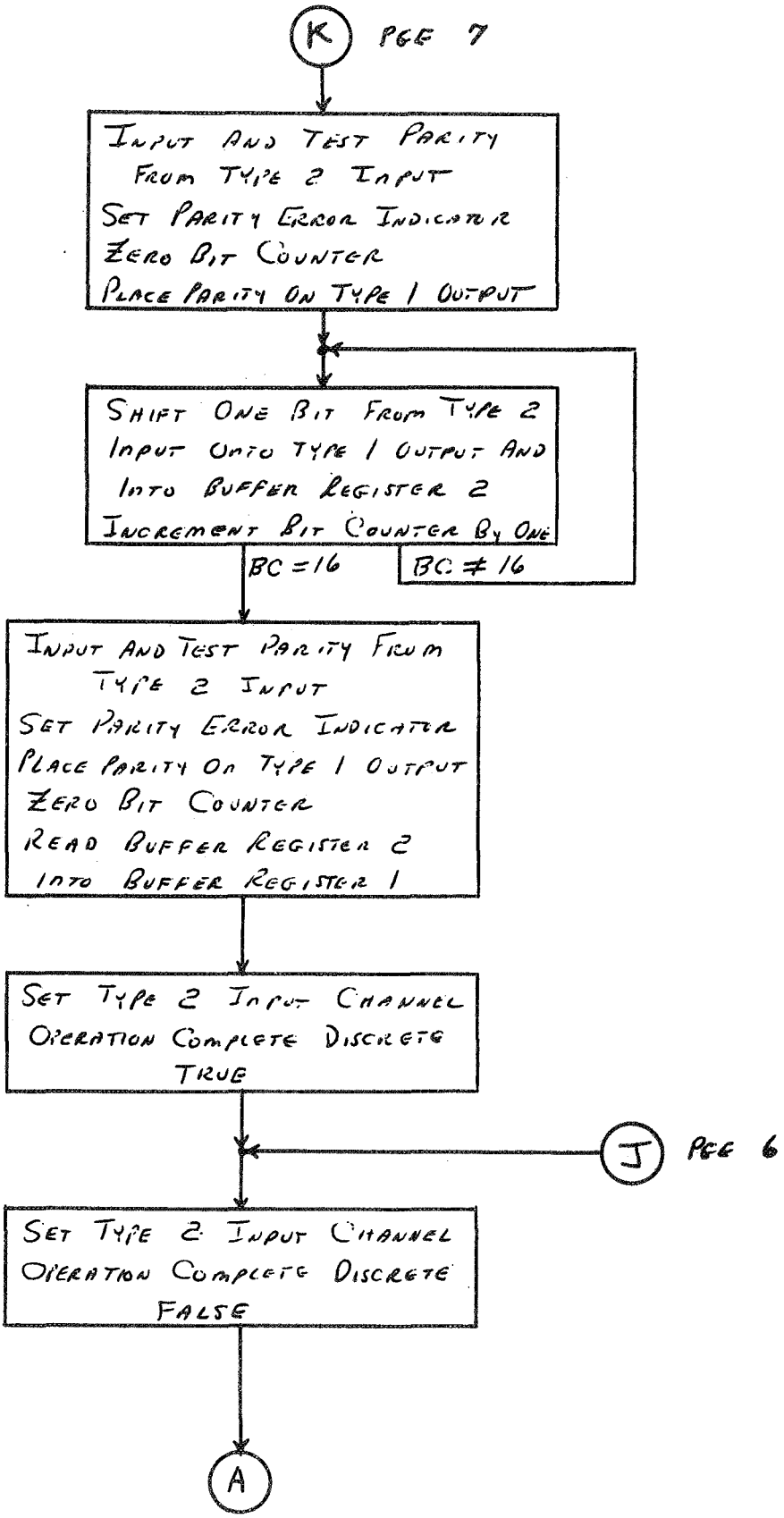

C
PGE 1
SHIFT ONE BIT FROM TYPE 2 INPUT
ONTO TYPE 1 OUTPUT
INCREMENT BIT COUNTER BY ONE
BC ≠ 4
BC = 4
SHIFT ONE BIT FROM TYPE 2 INPUT
ONTO TYPE 1 OUTPUT AND
INTO BUFFER REGISTER 2
COMPARE INPUT BIT WITH MSB
OF COMPARISON REGISTER
SET COMPARISON INDICATOR
INCREMENT BIT COUNTER BY ONE
BC = 20
BC ≠ 20
INPUT AND TEST PARITY
FROM TYPE 2 INPUT
SET PARITY ERROR INDICATOR
PLACE PARITY ON TYPE 1 OUTPUT
ZERO BIT COUNTER
COMPARE INPUT BIT WITH MSB
OF COMPARISON REGISTER
SET COMPARISON INDICATOR
SHIFT ONE BIT FROM TYPE 2
INPUT ONTO TYPE 1 OUTPUT AND
INTO BUFFER REGISTER 2
COMPARE INPUT BIT WITH MSB
OF COMPARISON REGISTER
SET COMPARISON INDICATOR
INCREMENT BIT COUNTER BY ONE
(BC=16)
(INPUT)
(BC=16)
(OUTPUT)
(OWN VCS)
BC ≠ 16
D
Pge 4
E
Pge 7


D
PGE 3
INPUT AND TEST PARITY
FROM TYPE 2 INPUT
SET PARITY ERROR INDICATOR
PLACE PARITY ON TYPE 1 OUTPUT
ZERO BIT COUNTER
READ BUFFER REGISTER 2
INTO BUFFER REGISTER 1
SHIFT ONE BIT FROM TYPE 2
INPUT ONTO TYPE 1 OUTPUT
AND INTO BUFFER REGISTER 2
INCREMENT BIT COUNTER BY ONE
READ FROM BUFFER REG 1 INTO
ACKNOWLEDGE WORD STORE:
1. BITS 7→16 INTO BITS 1→10
2. " 21→25 " " 11→16
READ BITS 1→6 OF BUFFER
REG 1 INTO WORD COUNTER
SHIFT ONE BIT FROM TYPE 2
INPUT ONTO TYPE 1 OUTPUT
AND INTO BUFFER REGISTER 2
INCREMENT BIT COUNTER BY ONE
COMPARE ACKNOWLEDGE WORD
STORE WITH CONTROL WORD REGISTER:
1. BITS 5→10 WITH BITS 21→25
2. " 11→16 " " 11→16
SET ACKNOWLEDGE WORD ERROR
INDICATOR AND ACKNOWLEDGE
WORD INDICATOR ACCORDINGLY
F
PGE 5

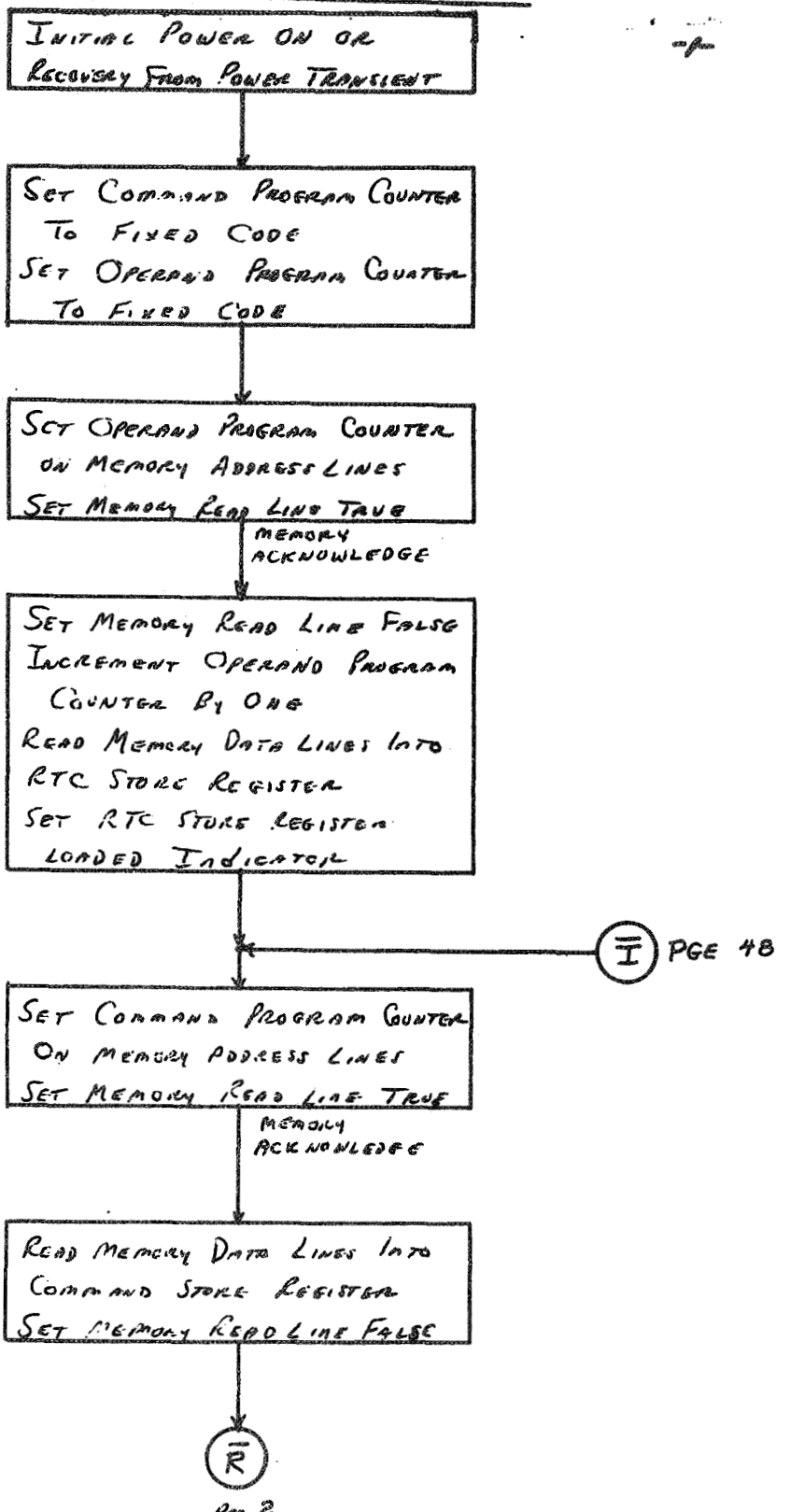

F
Pge 4
Shift One Bit From Type 2 Input Onto Type 1 Output And Into Buffer Register 2
Increment Bit Counter By One
BC ≠ 16
BC = 16
Input And Test Parity From Type 2 Input
Set Parity Error Indicator
Place Parity On Type 1 Output
Zero Bit Counter
Read Buffer Register 2 Into Buffer Register 1
Decrement Word Counter By One
WC ≠ 0
WC = 0
Pge 6
H
Pge 6
G
Pge 6
Shift One Bit From Type 2 Input Onto Type 1 Output And Into Buffer Register 2
Increment Bit Counter By One
Set Operand Program Counter On Memory Address Lines
Set Buffer Register 1 On Memory Data Lines
Set Memory Write Line True
Shift One Bit From Type 2 Input Onto Type 1 Output And Into Buffer Register 2
Increment Bit Counter By One
Test Memory Acknowledge Line
TRUE
FALSE
I
Pge 6

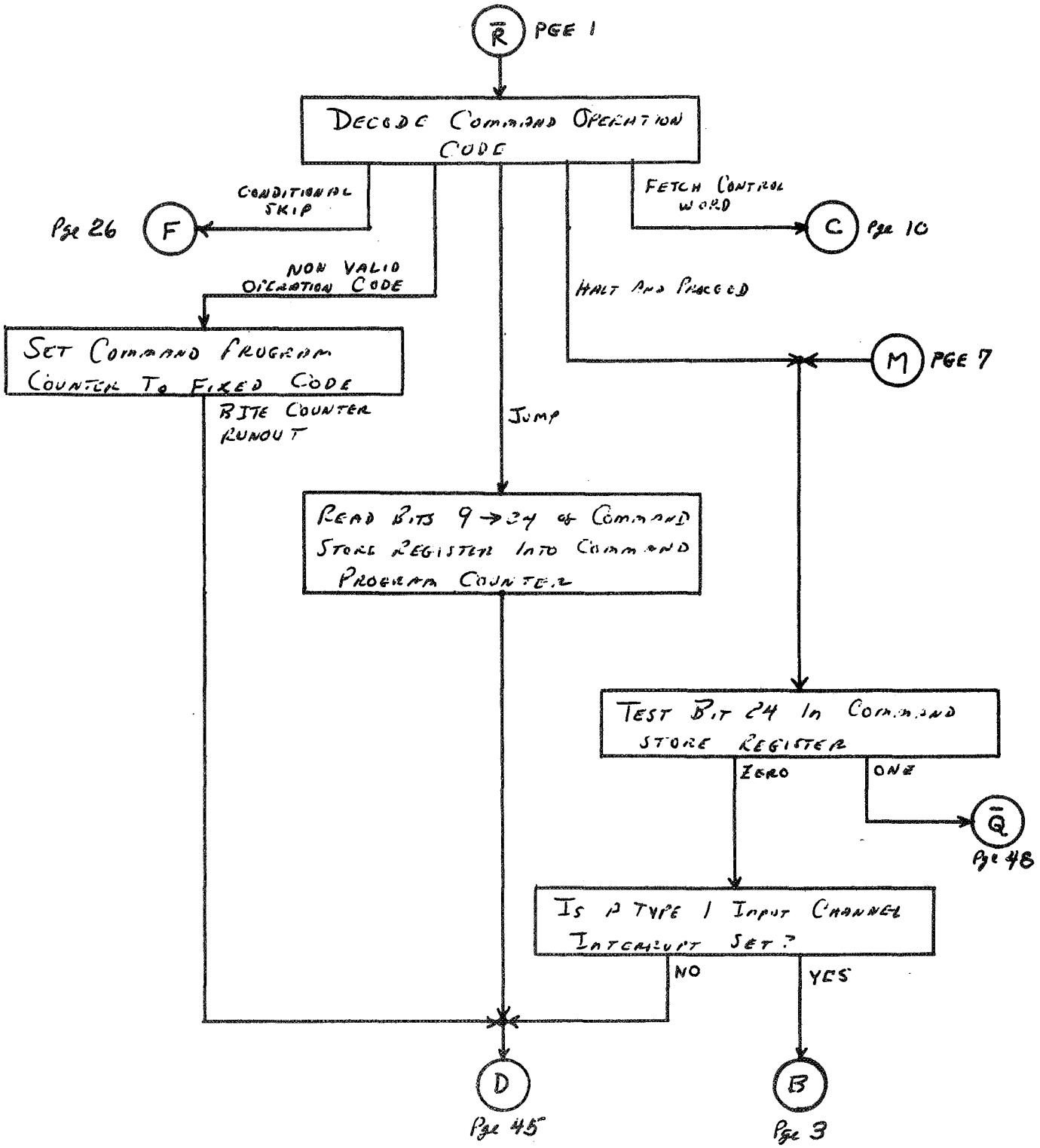

I
Pge 5
Shift One Bit From Type 2 Input Onto Type 1 Output And Into Buffer Register 2
Increment Bit Counter By One
Set Memory Write Line False
Increment Operand Program Counter By One
Shift One Bit From Type 2 Input Onto Type 1 Output And Into Buffer Register 2
Increment Bit Counter By One
BC ≠ 16
BC = 16
Input And Test Parity From Type 2 Input
Set Parity Error Indicator
Zero Bit Counter
Set Parity On Type 1 Output
Decrement Word Counter By One
Read Buffer Register 2 Into Buffer Register 1
WC = 0
WC ≠ 0
Pge 5
G
Pge 5
H
Pge 5
Set Operand Program Counter On Memory Address Lines
Set Buffer Register 1 On Memory Data Lines
Set Memory Write Line True
Memory Acknowledge
Increment Operand Program Counter By One
Set Memory Write Line False
Set Type 2 Input Channel Operation Complete Discrete True
J
Pge 8

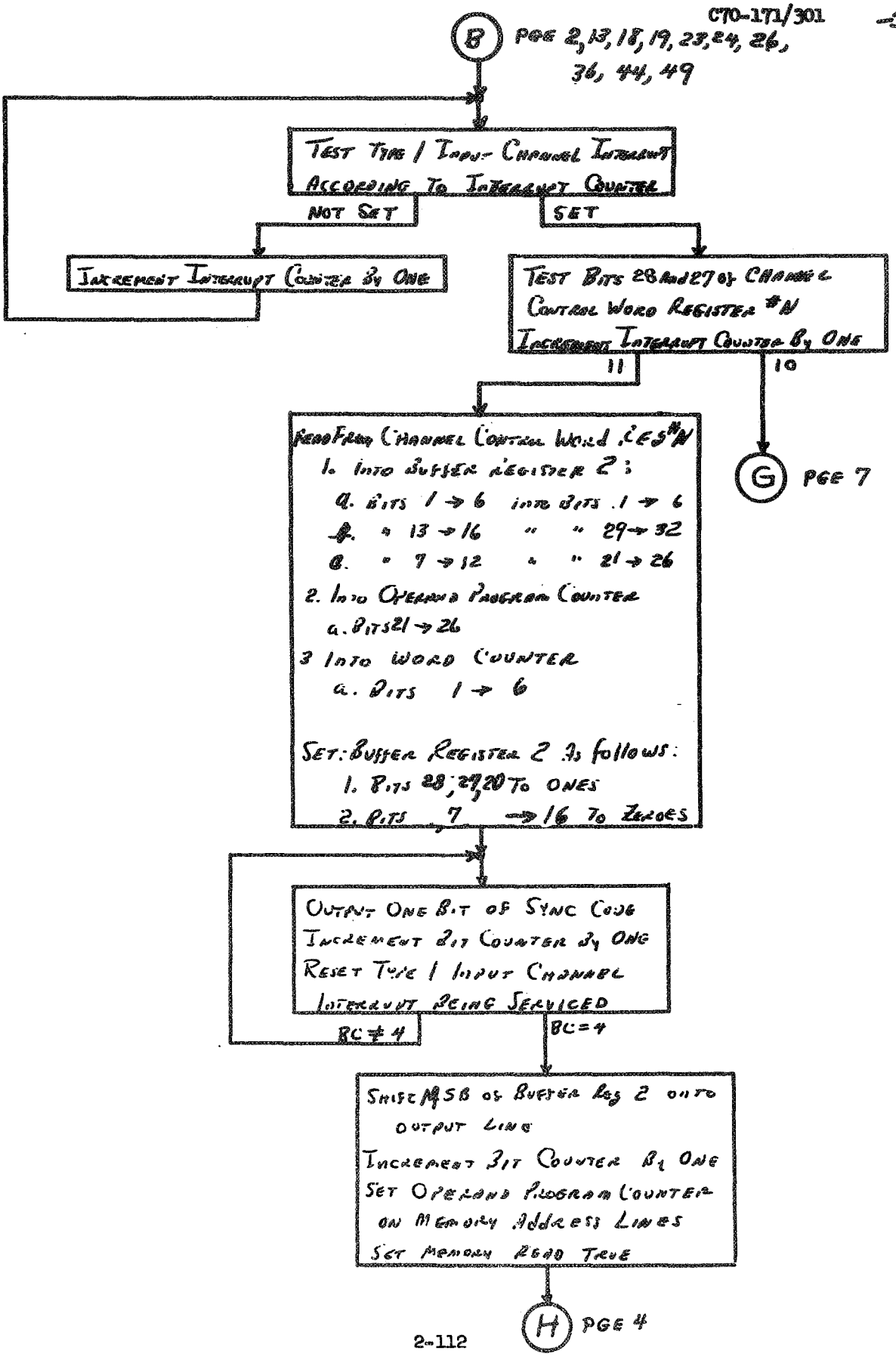

E
Pge 3
Input And Test Parity
From Type 2 Input
Set Parity Error Indicator
Zero Bit Counter
Compare Input Bit With MSB
of Comparison Register
Set Comparison Indicator
Shift One Bit From Type 2
Input Onto Type 1 Output
Increment Bit Counter By One
Compare Input Bit With MSB
of Comparison Register
Set Comparison Indicator
BC ≠ 16
BC = 16
Input And Test Parity
From Type 2 Input
Set Parity Error Indicator
Zero Bit Counter
Compare Input Bit With MSB
of Comparison Register
Set Comparison Indicator
Signal Present
Signal Absent
Test For Signal On Type 2 Input
Present
None
Shift One Bit From Type 2
Input Onto Type 1 Output
Increment Bit Counter By One
BC ≠ 4
BC = 4
Shift One Bit From Type 2
Input Onto Type 1 Output
And Into Buffer Register 2
Increment Bit Counter By One
BC = 20
BC ≠ 20
Pge 8
K

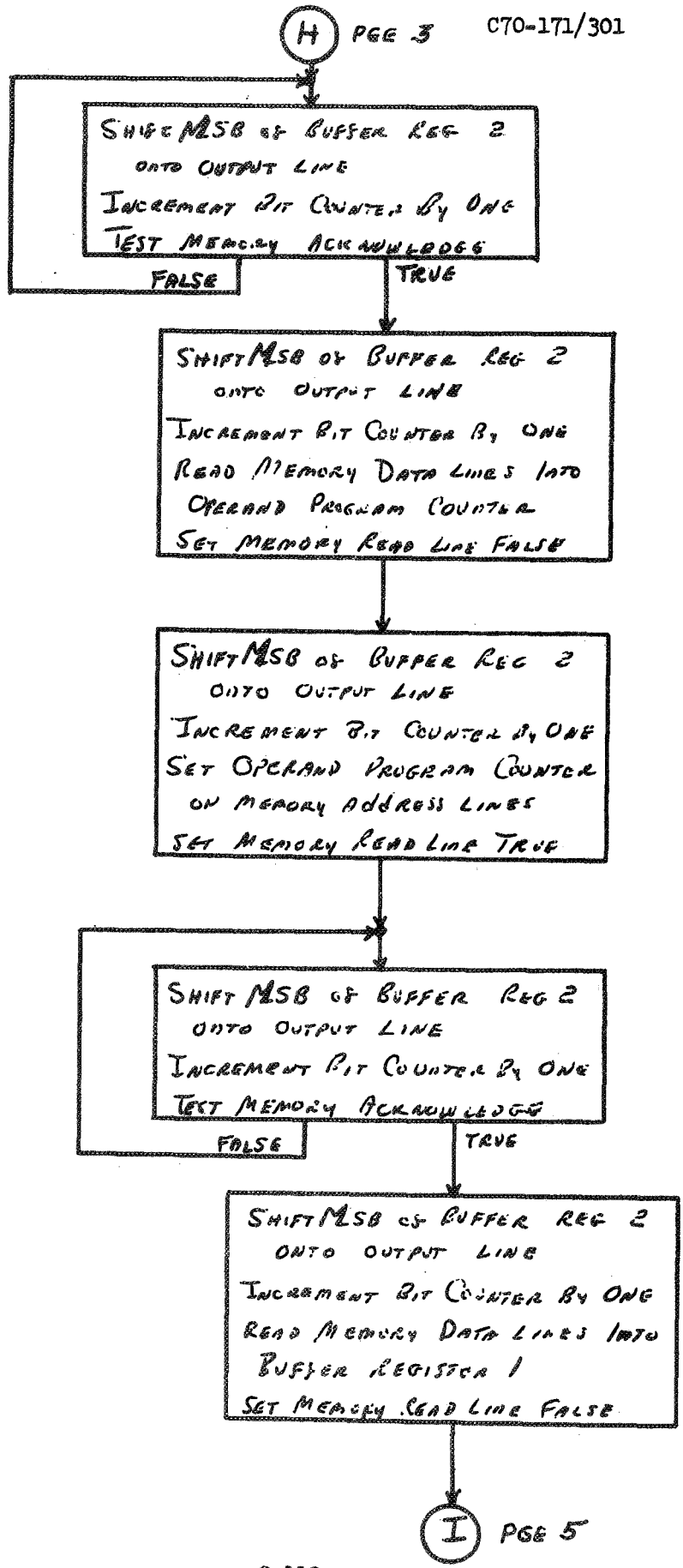

K
PGE 7
INPUT AND TEST PARITY FROM TYPE 2 INPUT
SET PARITY ERROR INDICATOR
ZERO BIT COUNTER
PLACE PARITY ON TYPE 1 OUTPUT
SHIFT ONE BIT FROM TYPE 2 INPUT ONTO TYPE 1 OUTPUT AND INTO BUFFER REGISTER 2
INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
INPUT AND TEST PARITY FROM TYPE 2 INPUT
SET PARITY ERROR INDICATOR
PLACE PARITY ON TYPE 1 OUTPUT
ZERO BIT COUNTER
READ BUFFER REGISTER 2 INTO BUFFER REGISTER 1
SET TYPE 2 INPUT CHANNEL OPERATION COMPLETE DISCRETE TRUE
J
PGE 6
SET TYPE 2 INPUT CHANNEL OPERATION COMPLETE DISCRETE FALSE
A
Pge 1

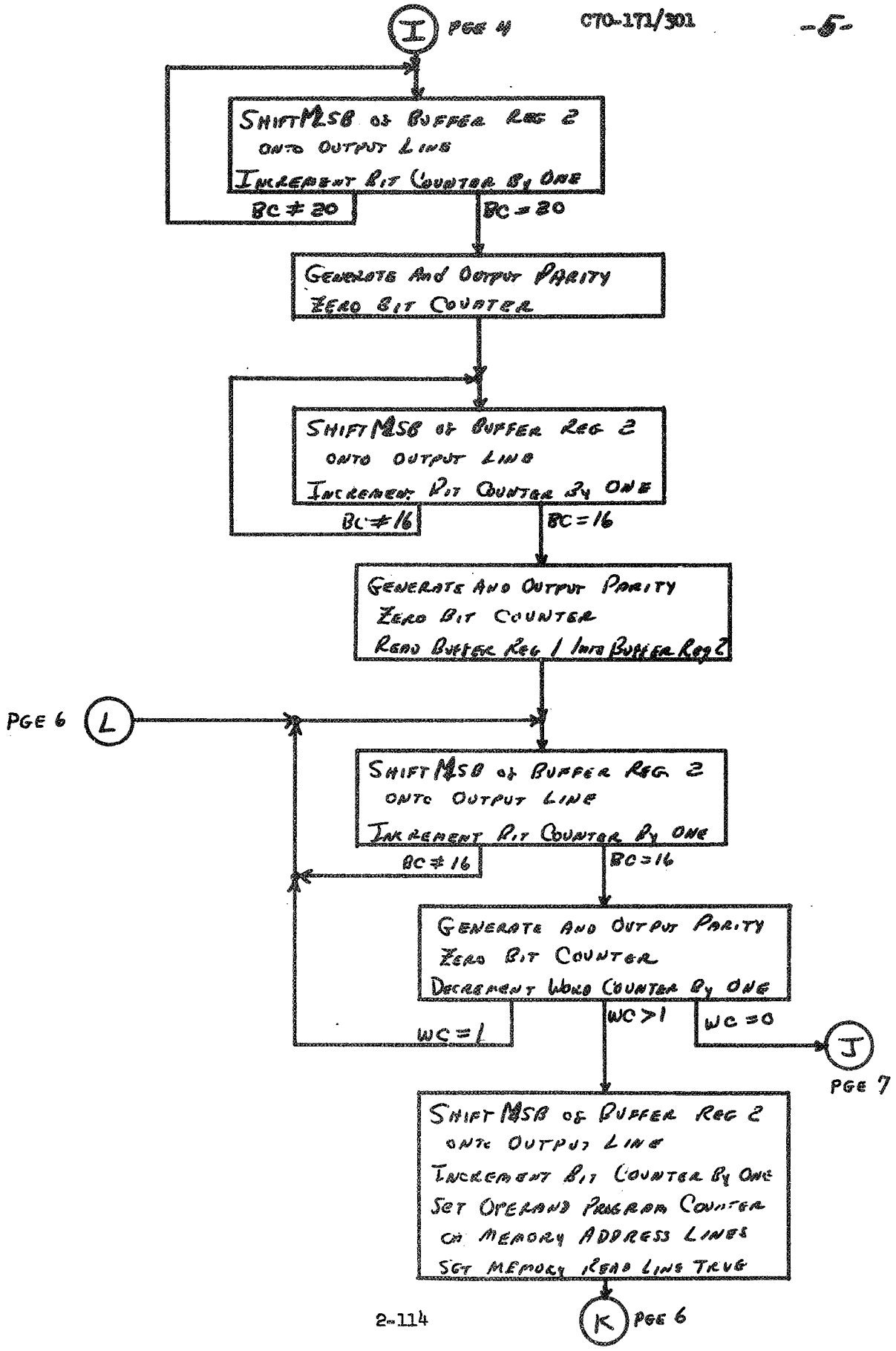
4.2.6 **Type 1 and 2 Output Channels and IOP Command Control**

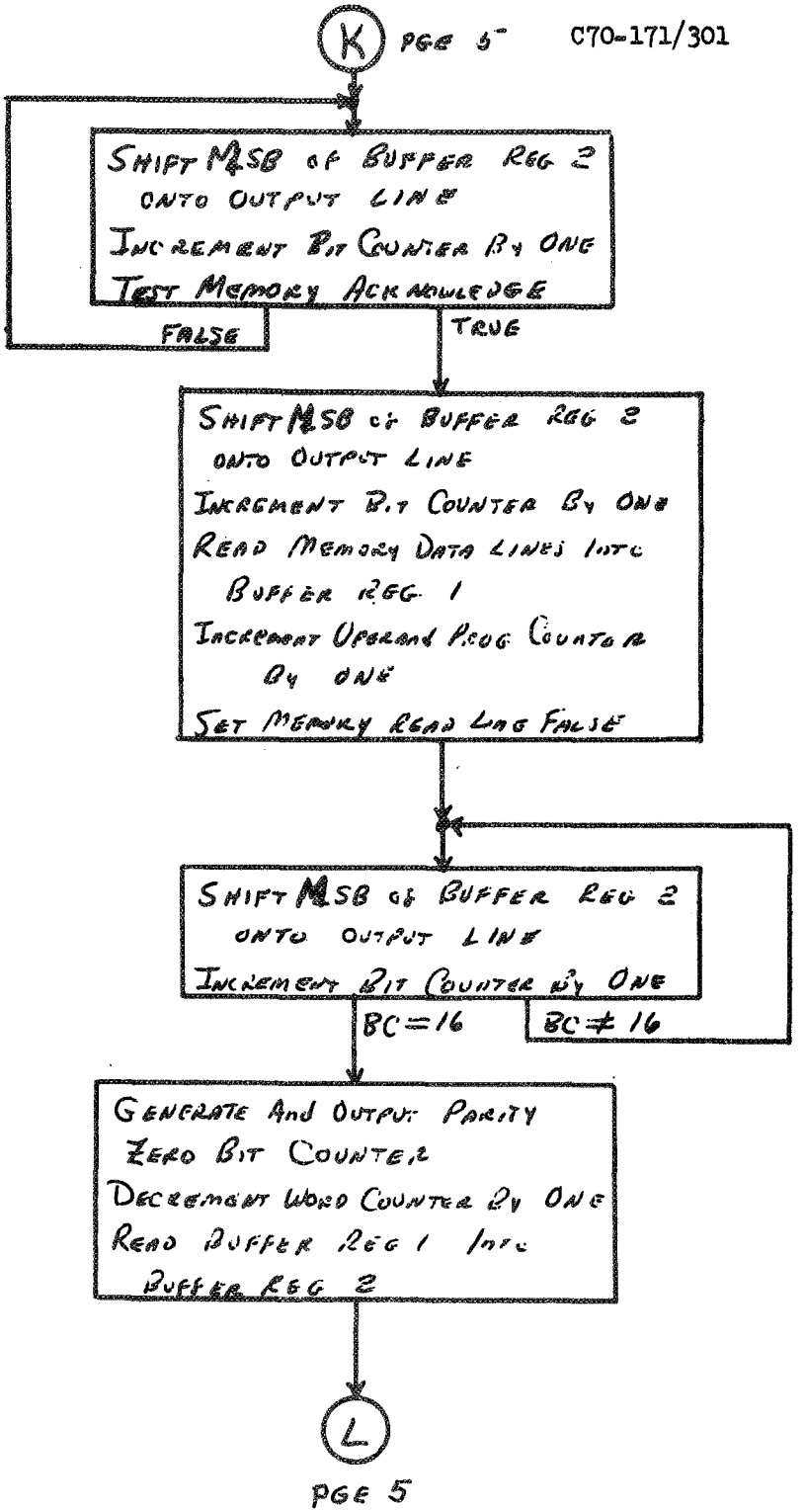


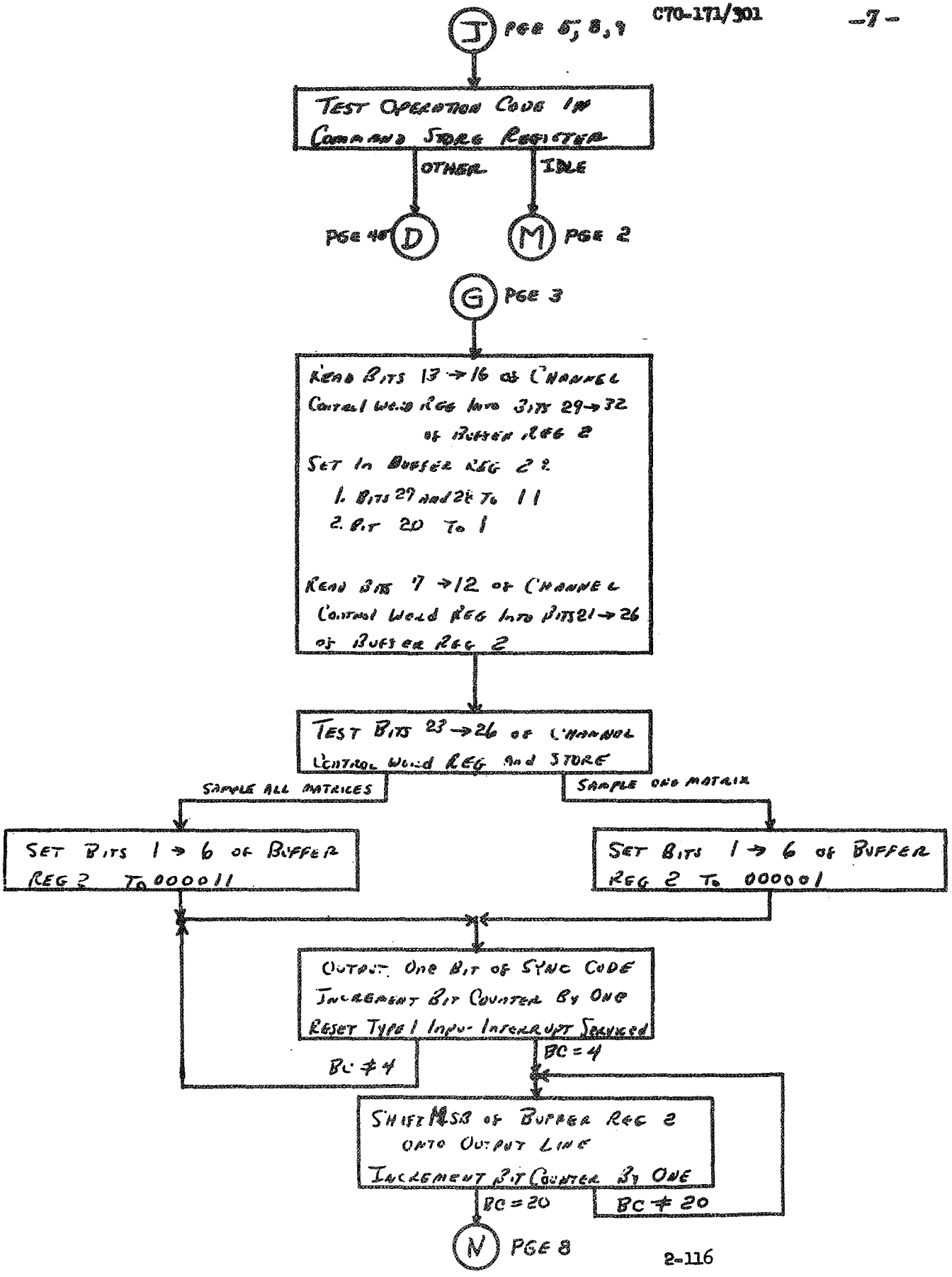

R
PGE 1
DECODE COMMAND OPERATION CODE
CONDITIONAL SKIP
Pge 26
F
FETCH CONTROL WORD
C
Pge 10
NON VALID OPERATION CODE
HALT AND PROCEED
SET COMMAND PROGRAM COUNTER TO FIXED CODE
M
PGE 7
BITE COUNTER RUNOUT
JUMP
READ BITS 9 → 24 of COMMAND STORE REGISTER INTO COMMAND PROGRAM COUNTER
TEST BIT 24 IN COMMAND STORE REGISTER
ZERO
ONE
Q
Pge 48
IS A TYPE I INPUT CHANNEL INTERRUPT SET?
NO
YES
D
Pge 45
B
Pge 3

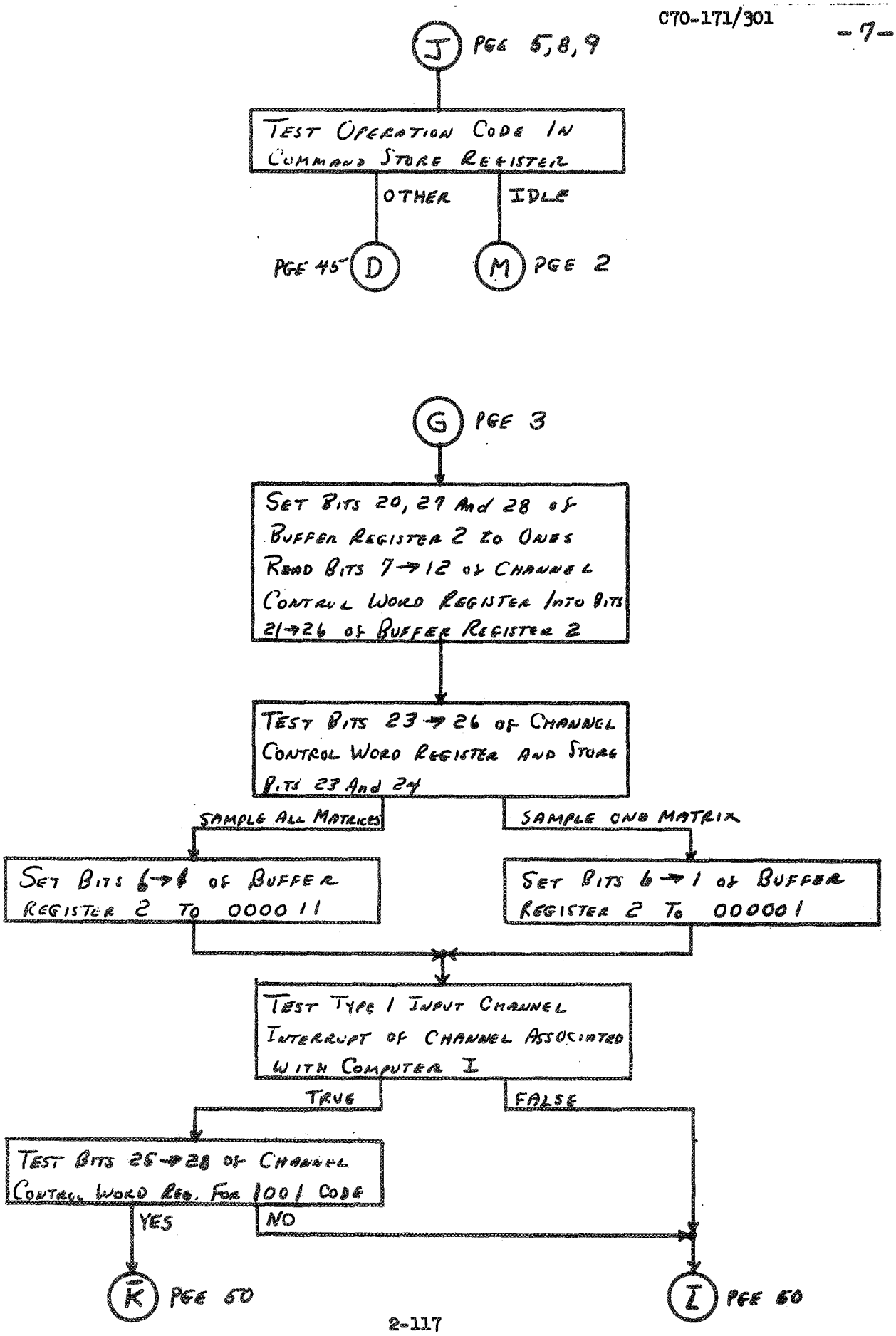

B
PGE 2, 13, 18, 19, 23, 24, 26, 36, 44, 49
TEST TYPE 1 INPUT CHANNEL INTERRUPT ACCORDING TO INTERRUPT COUNTER
NOT SET
SET
INCREMENT INTERRUPT COUNTER BY ONE
TEST BITS 28 AND 27 OF CHANNEL CONTROL WORD REGISTER #N
INCREMENT INTERRUPT COUNTER BY ONE
11
10
G
PGE 7
READ FROM CHANNEL CONTROL WORD REG #N
1. INTO BUFFER REGISTER 2:
a. BITS 1 → 6 INTO BITS 1 → 6
b. " 13 → 16 " " 29 → 32
c. " 7 → 12 " " 21 → 26
2. INTO OPERAND PROGRAM COUNTER
a. BITS 21 → 26
3 INTO WORD COUNTER
a. BITS 1 → 6
SET BUFFER REGISTER 2 AS FOLLOWS:
1. BITS 28, 27, 20 TO ONES
2. BITS 7 → 16 TO ZEROES
OUTPUT ONE BIT OF SYNC CODE
INCREMENT BIT COUNTER BY ONE
RESET TYPE 1 INPUT CHANNEL INTERRUPT BEING SERVICED
BC ≠ 4
BC = 4
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ TRUE
H
PGE 4

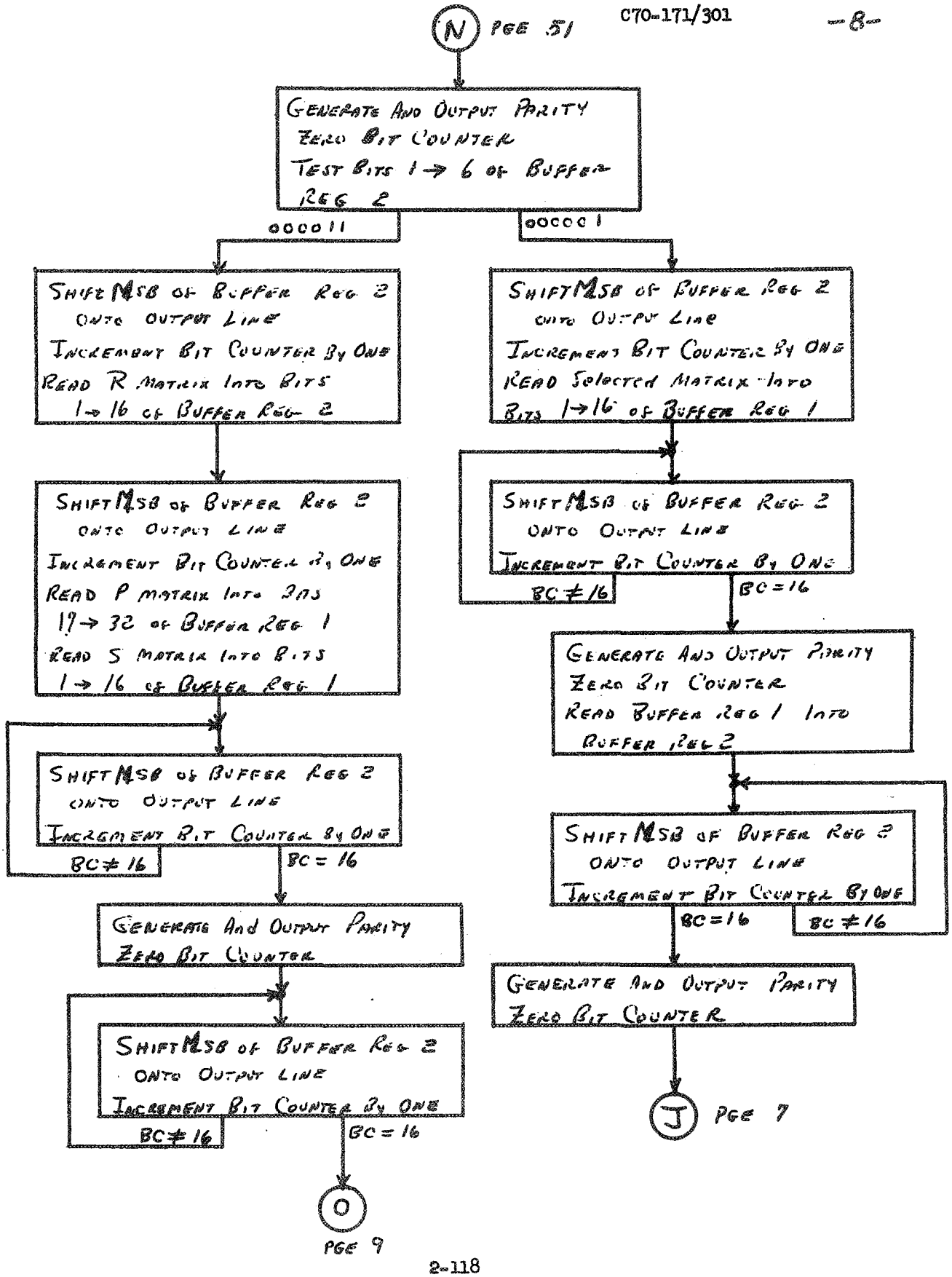

H
PGE 3
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO
OPERAND PROGRAM COUNTER
SET MEMORY READ LINE FALSE
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER
ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO
BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
I
PGE 5

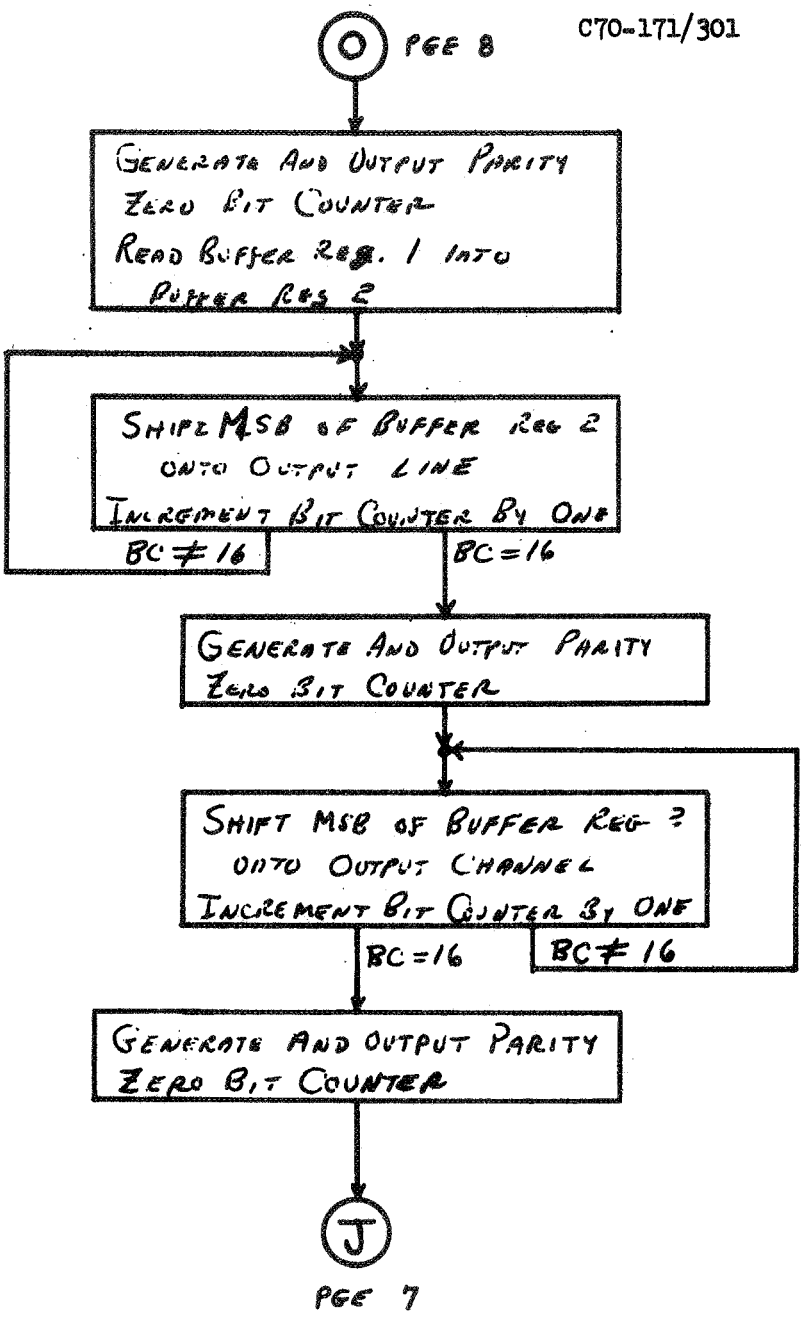

I
PGE 4
SHIFT MSB of BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 20
BC = 20
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
SHIFT MSB of BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
READ BUFFER REG 1 INTO BUFFER REG 2
PGE 6
L
SHIFT MSB of BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
WC = 1
WC > 1
WC = 0
J
PGE 7
SHIFT MSB of BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
K
PGE 6

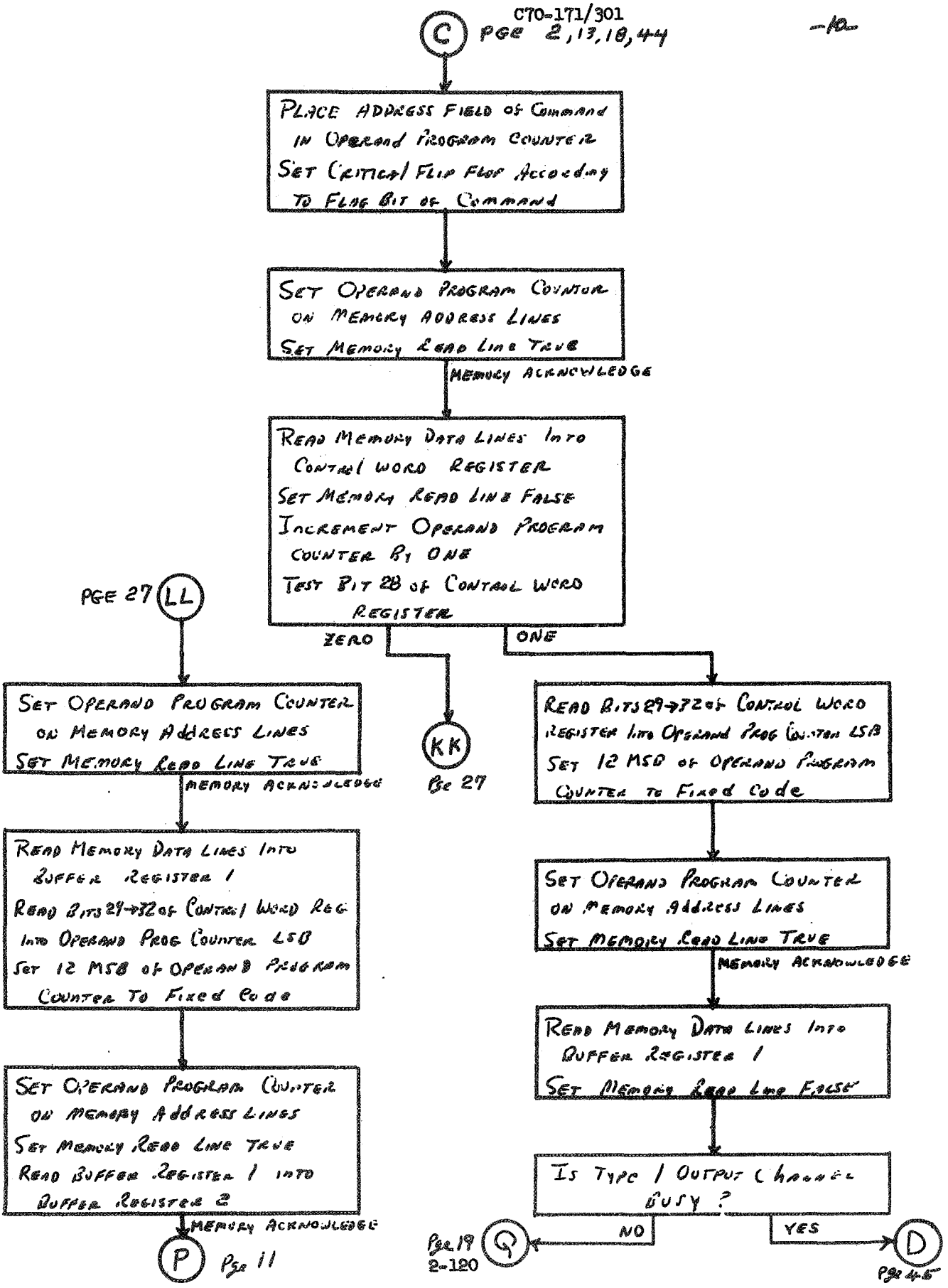

K
PGE 5
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO
BUFFER REG 1
INCREMENT OPERAND PROG COUNTER
BY ONE
SET MEMORY READ LINE FALSE
SHIFT MSB of BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=16
BC≠16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
READ BUFFER REG 1 INTO
BUFFER REG 2
L
PGE 5

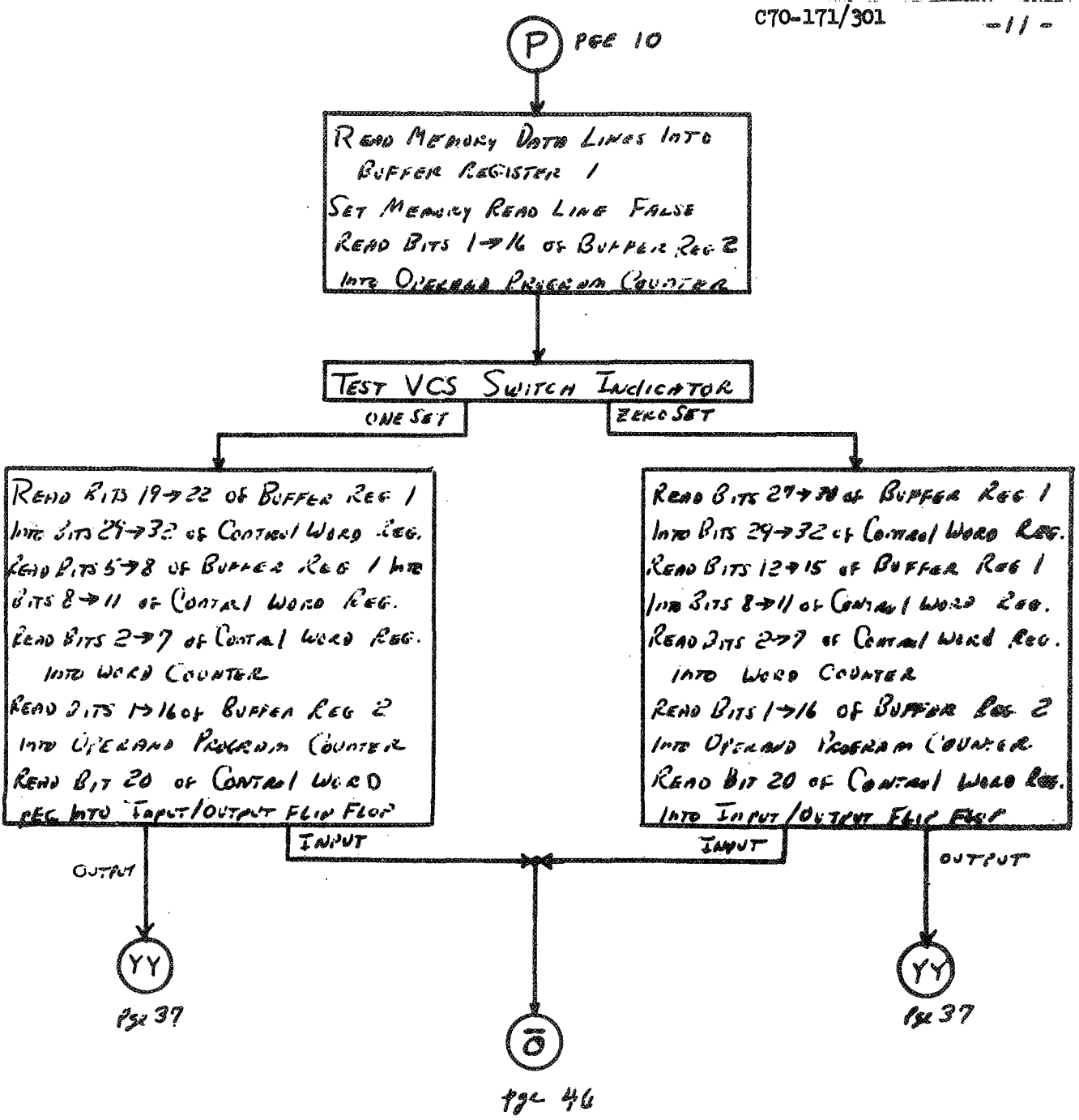

J PGE 6, 8, 9
TEST OPERATION CODE IN COMMAND STORE REGISTER
OTHER
IDLE
PGE 46 D
M PGE 2
G PGE 3
READ BITS 13 → 16 OF CHANNEL CONTROL WORD REG INTO BITS 29 → 32 OF BUFFER REG 2
SET IN BUFFER REG 2:
1. BITS 29 AND 28 TO 11
2. BIT 20 TO 1
READ BITS 7 → 12 OF CHANNEL CONTROL WORD REG INTO BITS 21 → 26 OF BUFFER REG 2
TEST BITS 23 → 26 OF CHANNEL CONTROL WORD REG AND STORE
SAMPLE ALL MATRICES
SAMPLE ONE MATRIX
SET BITS 1 → 6 OF BUFFER REG 2 TO 000011
SET BITS 1 → 6 OF BUFFER REG 2 TO 000001
OUTPUT ONE BIT OF SYNC CODE
INCREMENT BIT COUNTER BY ONE
RESET TYPE I INPUT-INTERRUPT SERVICED
BC ≠ 4
BC = 4
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 20
BC ≠ 20
N PGE 8

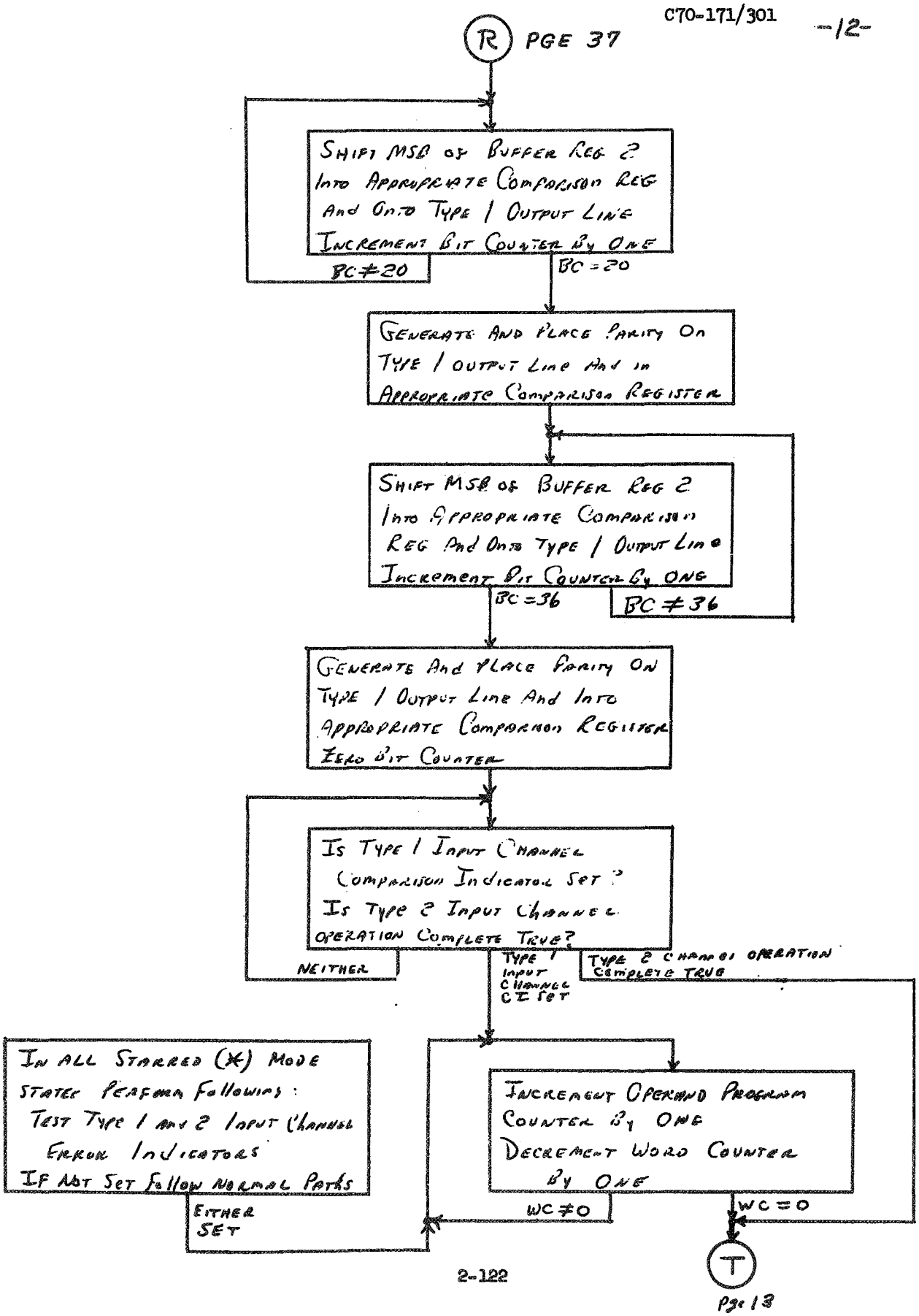

J PGE 5,8,9
TEST OPERATION CODE IN COMMAND STORE REGISTER
OTHER
IDLE
PGE 45 D
M PGE 2
G PGE 3
SET BITS 20, 27 And 28 of BUFFER REGISTER 2 to ONES READ BITS 7→12 of CHANNEL CONTROL WORD REGISTER INTO BITS 21→26 of BUFFER REGISTER 2
TEST BITS 23→26 of CHANNEL CONTROL WORD REGISTER AND STORE BITS 23 And 24
SAMPLE ALL MATRICES
SAMPLE ONE MATRIX
SET BITS 6→1 of BUFFER REGISTER 2 TO 000011
SET BITS 6→1 of BUFFER REGISTER 2 To 000001
TEST TYPE I INPUT CHANNEL INTERRUPT OF CHANNEL ASSOCIATED WITH COMPUTER I
TRUE
FALSE
TEST BITS 25→28 of CHANNEL CONTROL WORD REG. FOR 1001 CODE
YES
NO
K PGE 50
L PGE 60

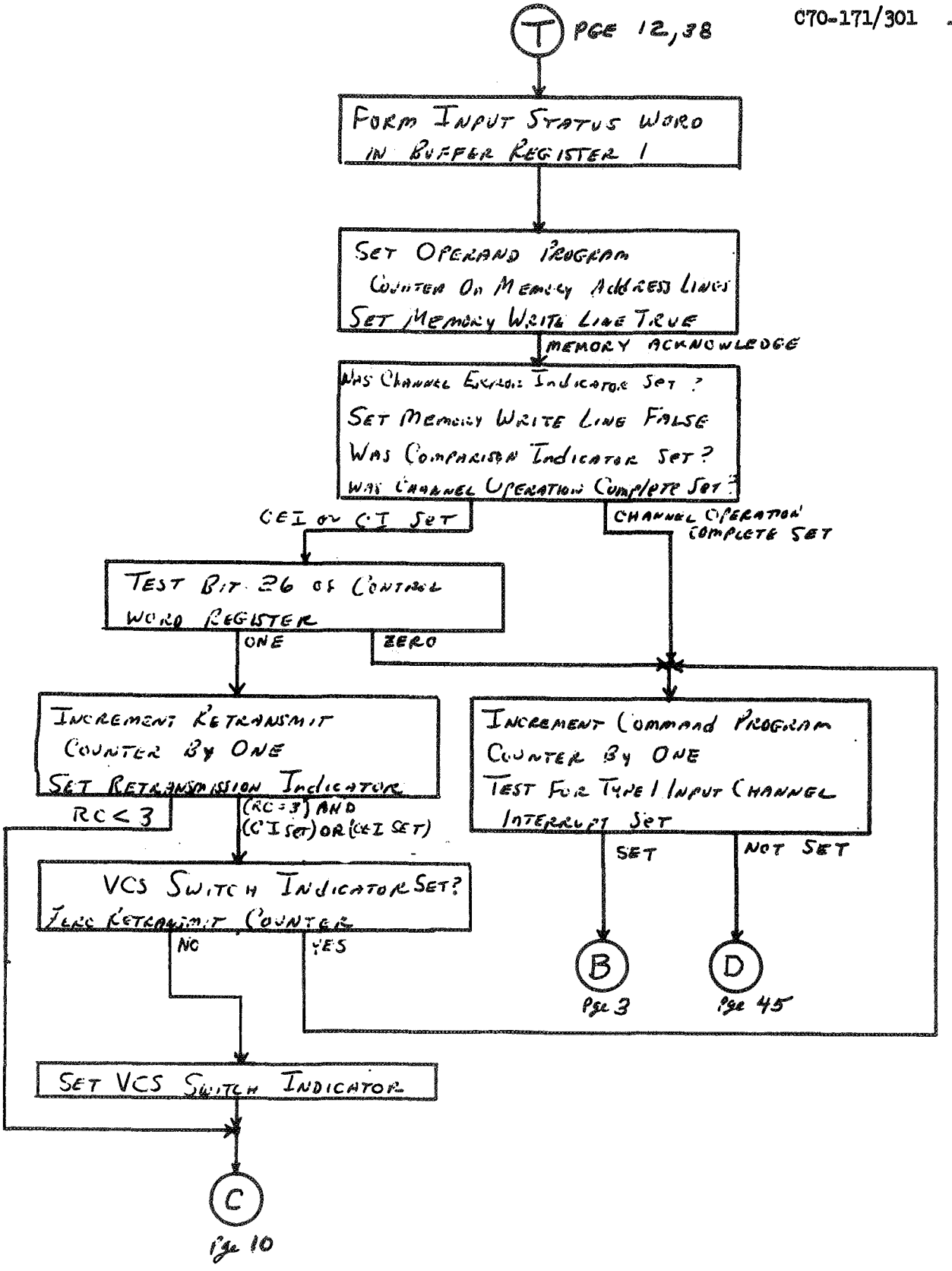

N PGE 51
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
TEST BITS 1 → 6 OF BUFFER REG 2
000011
000001
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ R MATRIX INTO BITS 1 → 16 OF BUFFER REG 2
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ P MATRIX INTO BITS 17 → 32 OF BUFFER REG 1
READ S MATRIX INTO BITS 1 → 16 OF BUFFER REG 1
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
O PGE 9
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ SELECTED MATRIX INTO BITS 1 → 16 OF BUFFER REG 1
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
READ BUFFER REG 1 INTO BUFFER REG 2
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
J PGE 7

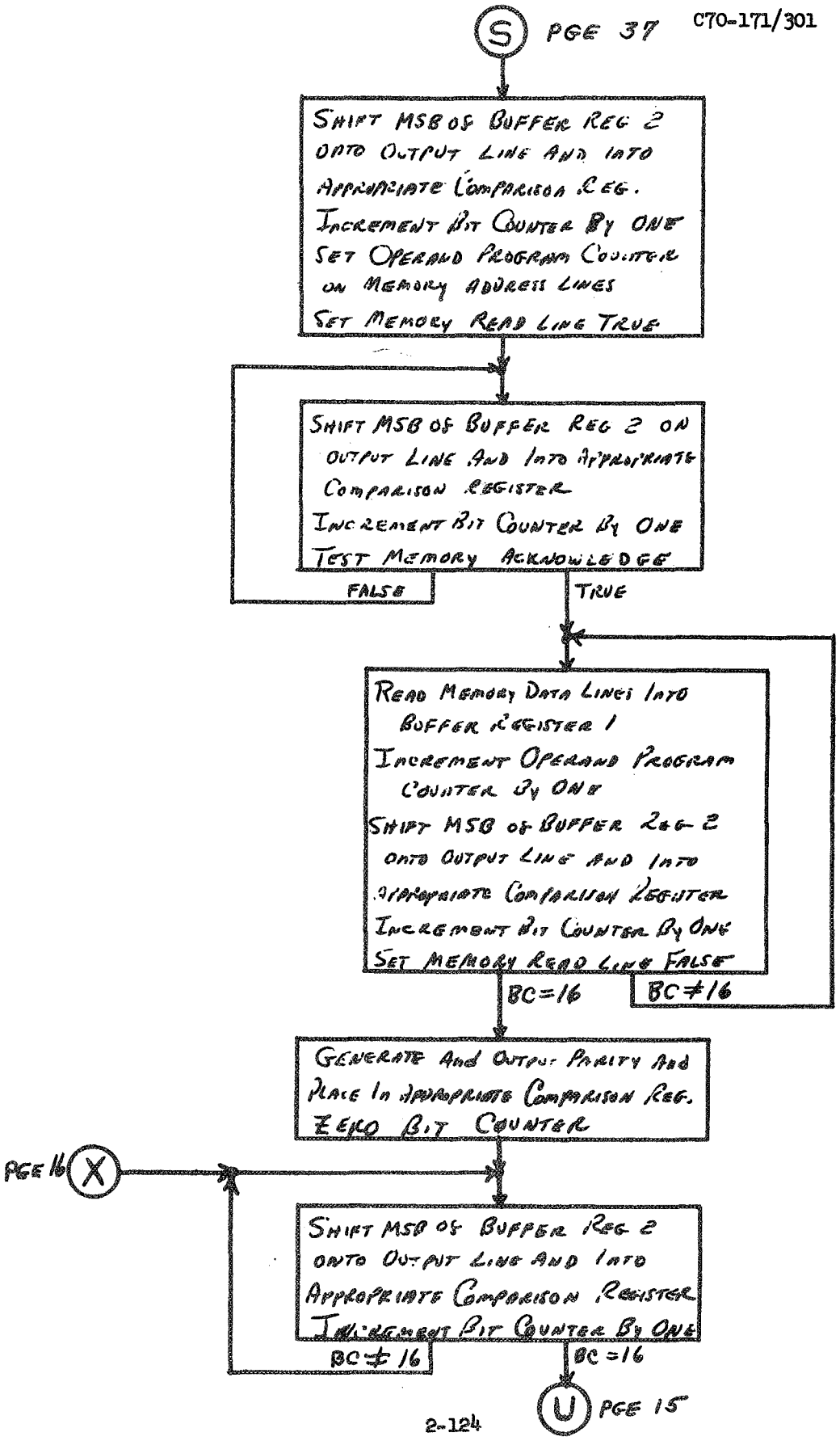

O
PGE 8
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
READ BUFFER REG. 1 INTO
BUFFER REG 2
SHIFT MSB OF BUFFER REG 2
ONTO OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
SHIFT MSB OF BUFFER REG 2
ONTO OUTPUT CHANNEL
INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
GENERATE AND OUTPUT PARITY
ZERO BIT COUNTER
J
PGE 7

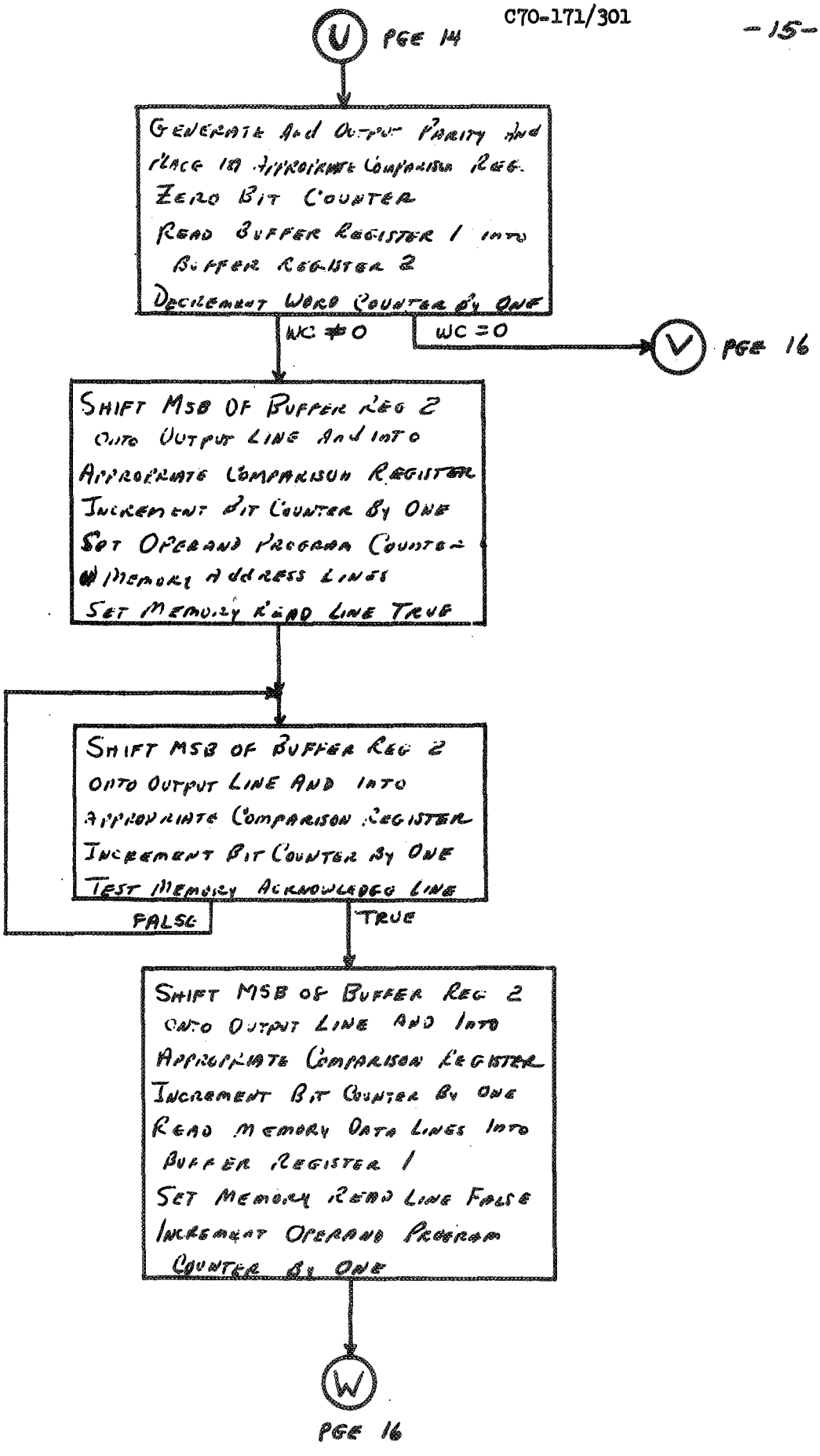

C
PGE 2,13,18,44
PLACE ADDRESS FIELD of Command IN Operand Program Counter
SET Critical Flip Flop Accordng TO FLAG BIT of Command
SET Operand Program Counter ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES Into Control WORD REGISTER
SET MEMORY READ LINE FALSE
INCREMENT Operand PROGRAM COUNTER BY ONE
TEST BIT 28 of CONTROL WORD REGISTER
ZERO
ONE
PGE 27 LL
SET Operand PROGRAM COUNTER ON MEMORY Address LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES Into BUFFER REGISTER 1
READ BITS 29→32 of Control WORD REG Into Operand PROG COUNTER LSB
SET 12 MSB of Operand PROGRAM COUNTER TO Fixed Code
SET Operand PROGRAM COUNTER ON MEMORY Address LINES
SET MEMORY READ LINE TRUE
READ BUFFER REGISTER 1 INTO BUFFER REGISTER 2
MEMORY ACKNOWLEDGE
P
Pge 11
KK
Pge 27
READ BITS 29→32 of CONTROL WORD REGISTER Into Operand PROG COUNTER LSB
SET 12 MSB of Operand PROGRAM COUNTER TO Fixed Code
SET Operand PROGRAM COUNTER ON MEMORY Address LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES Into BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
IS TYPE 1 OUTPUT CHANNEL BUSY ?
NO
YES
Pge 19
2-120
Q
D
Pge 4-5

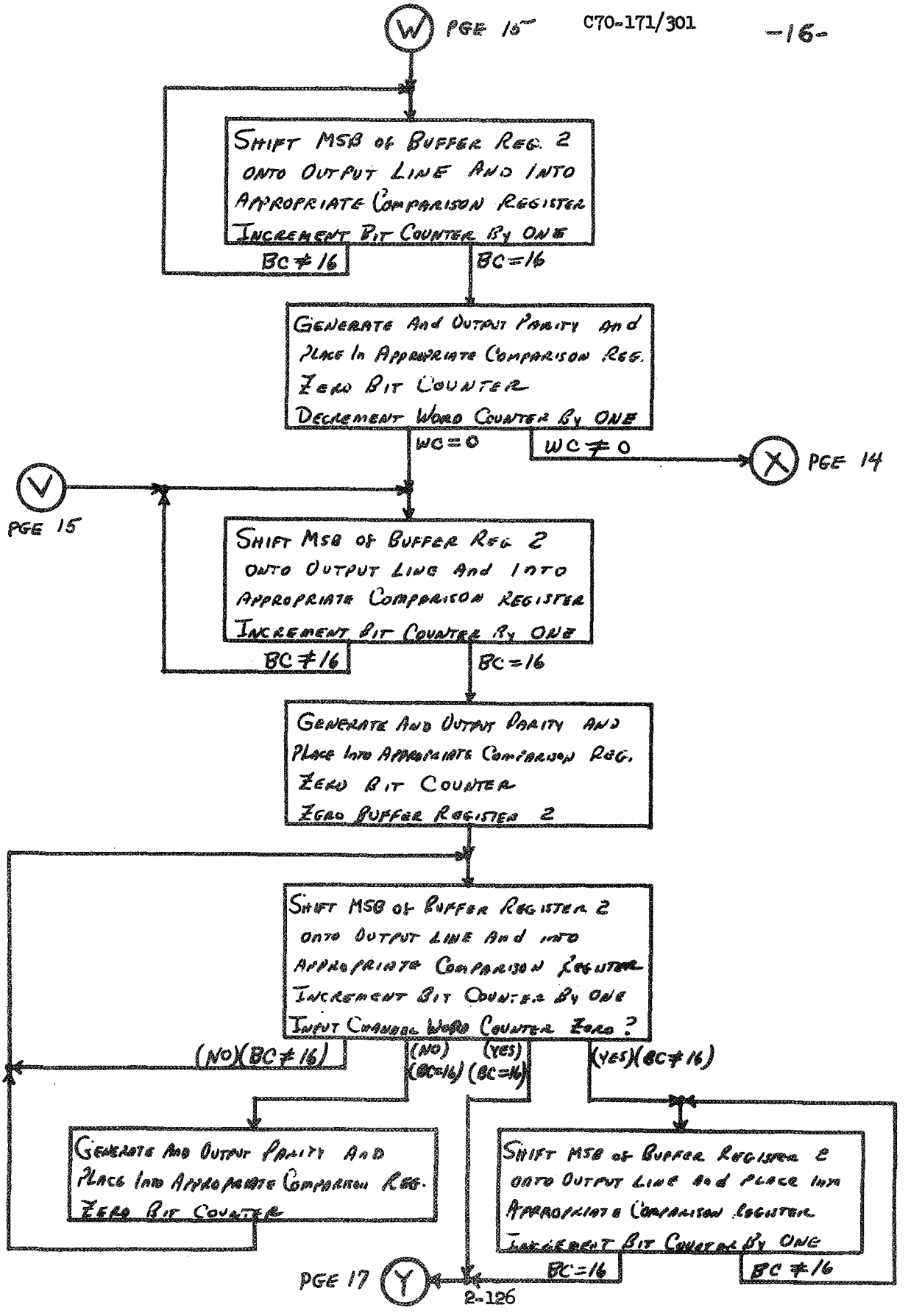

P
PGE 10
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
READ BITS 1→16 OF BUFFER REG 2 INTO OPERAND PROGRAM COUNTER
TEST VCS SWITCH INDICATOR
ONE SET
ZERO SET
READ BITS 19→22 OF BUFFER REG 1 INTO BITS 29→32 OF CONTROL WORD REG.
READ BITS 5→8 OF BUFFER REG 1 INTO BITS 8→11 OF CONTROL WORD REG.
READ BITS 2→7 OF CONTROL WORD REG. INTO WORD COUNTER
READ BITS 1→16 OF BUFFER REG 2 INTO OPERAND PROGRAM COUNTER
READ BIT 20 OF CONTROL WORD REG INTO INPUT/OUTPUT FLIP FLOP
READ BITS 27→30 OF BUFFER REG 1 INTO BITS 29→32 OF CONTROL WORD REG.
READ BITS 12→15 OF BUFFER REG 1 INTO BITS 8→11 OF CONTROL WORD REG.
READ BITS 2→7 OF CONTROL WORD REG. INTO WORD COUNTER
READ BITS 1→16 OF BUFFER REG 2 INTO OPERAND PROGRAM COUNTER
READ BIT 20 OF CONTROL WORD REG. INTO INPUT/OUTPUT FLIP FLOP
INPUT
INPUT
OUTPUT
OUTPUT
YY
PGE 37
YY
PGE 37
O
PGE 46

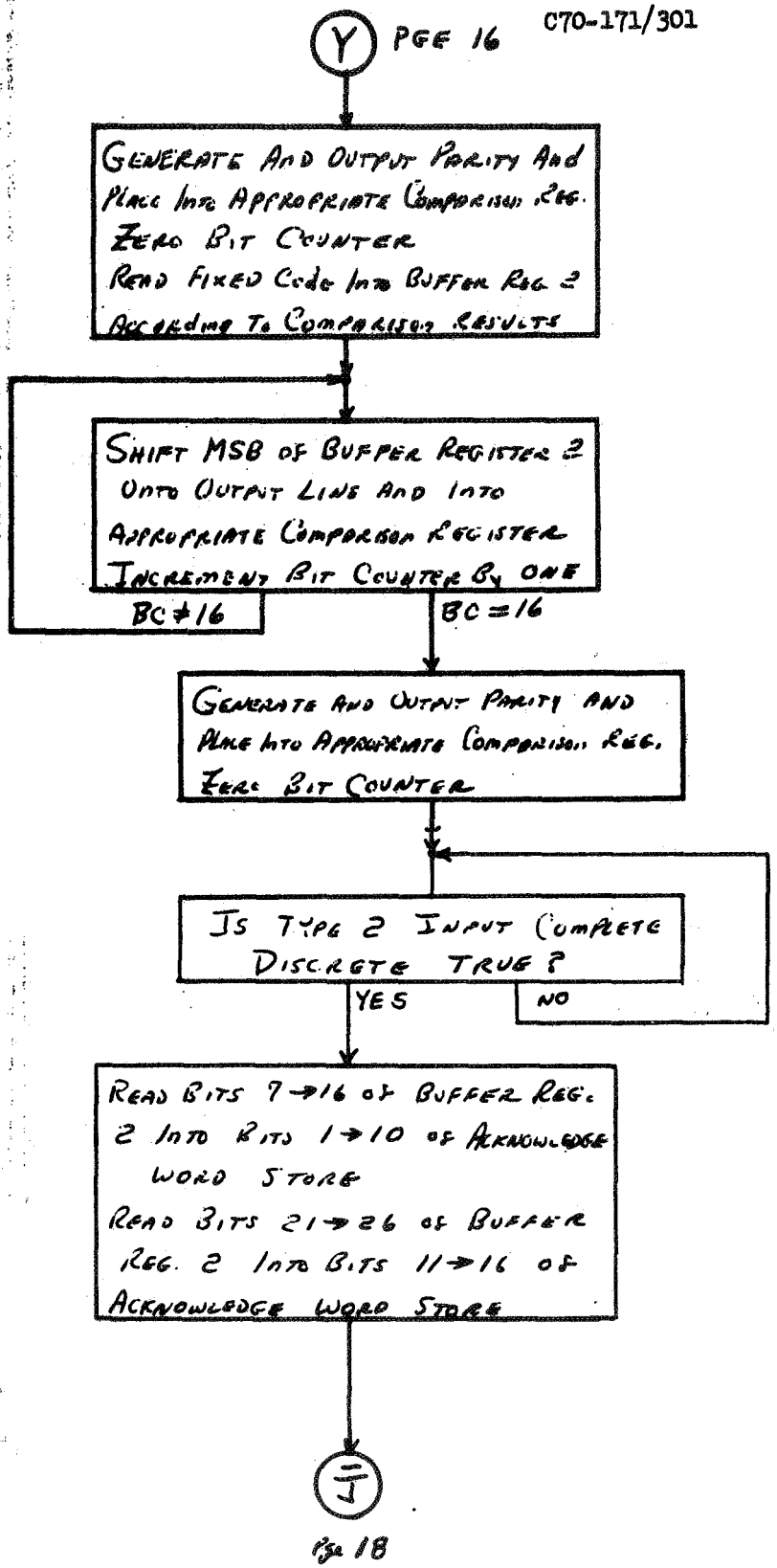

R
PGE 37
SHIFT MSB OF BUFFER REG 2 INTO APPROPRIATE COMPARISON REG AND ONTO TYPE / OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 20
BC = 20
GENERATE AND PLACE PARITY ON TYPE / OUTPUT LINE AND IN APPROPRIATE COMPARISON REGISTER
SHIFT MSB OF BUFFER REG 2 INTO APPROPRIATE COMPARISON REG AND ONTO TYPE / OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 36
BC ≠ 36
GENERATE AND PLACE PARITY ON TYPE / OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
ZERO BIT COUNTER
IS TYPE 1 INPUT CHANNEL COMPARISON INDICATOR SET ?
IS TYPE 2 INPUT CHANNEL OPERATION COMPLETE TRUE?
NEITHER
TYPE 1 INPUT CHANNEL CE SET
TYPE 2 CHANNEL OPERATION COMPLETE TRUE
IN ALL STARRED (*) MODE STATES PERFORM FOLLOWING :
TEST TYPE 1 AND 2 INPUT CHANNEL ERROR INDICATORS
IF NOT SET FOLLOW NORMAL PATHS
EITHER SET
INCREMENT OPERAND PROGRAM COUNTER BY ONE
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
T
Pg. 13

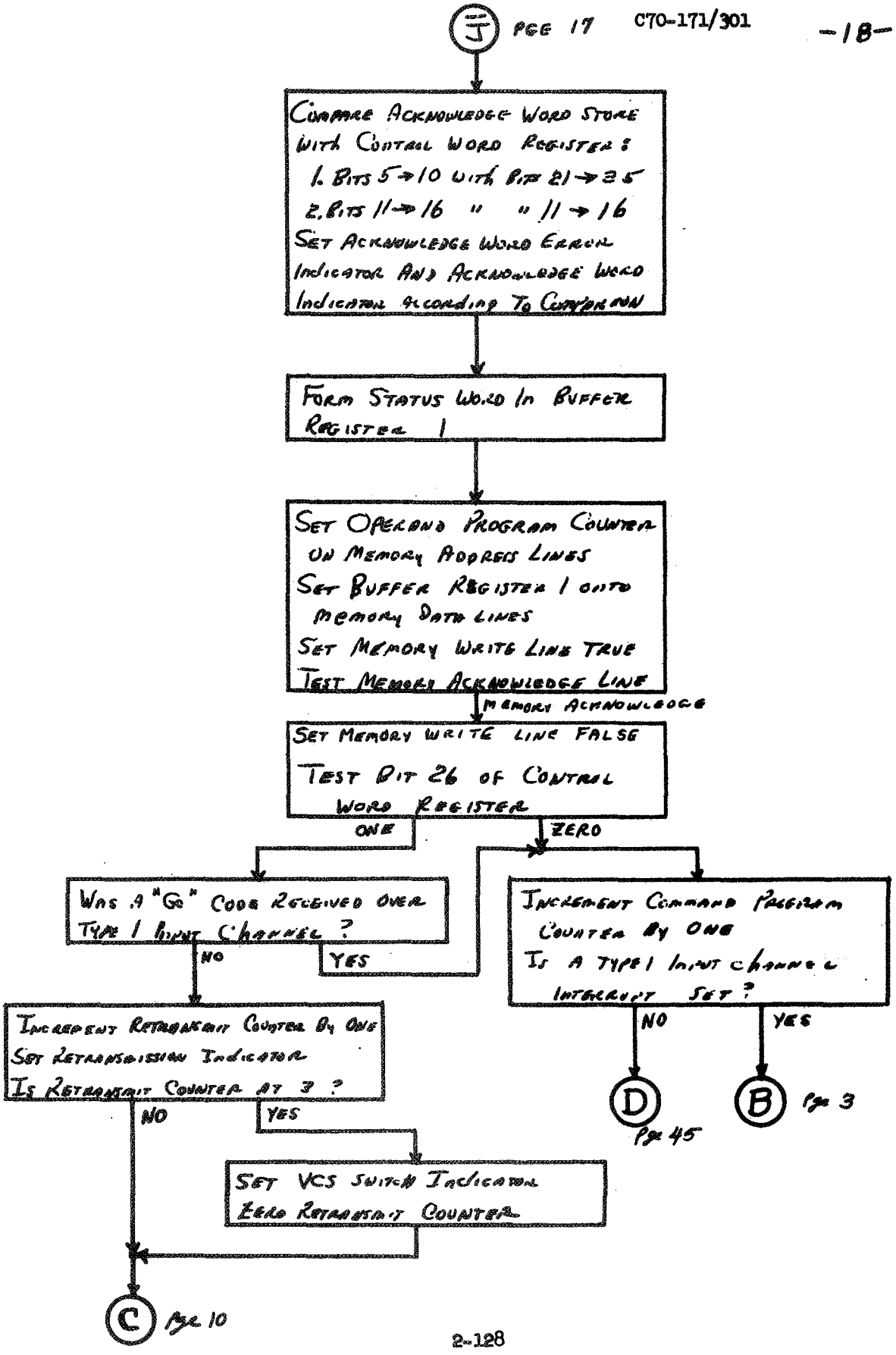

T
PGE 12,38
FORM INPUT STATUS WORD IN BUFFER REGISTER 1
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
WAS CHANNEL ERROR INDICATOR SET ?
SET MEMORY WRITE LINE FALSE
WAS COMPARISON INDICATOR SET?
WAS CHANNEL OPERATION COMPLETE SET ?
CEI OR CI SET
CHANNEL OPERATION COMPLETE SET
TEST BIT 26 OF CONTROL WORD REGISTER
ONE
ZERO
INCREMENT RETRANSMIT COUNTER BY ONE
SET RETRANSMISSION INDICATOR
INCREMENT COMMAND PROGRAM COUNTER BY ONE
TEST FOR TYPE I INPUT CHANNEL INTERRUPT SET
RC < 3
(RC = 3) AND ((CI SET) OR (CEI SET))
SET
NOT SET
VCS SWITCH INDICATOR SET?
ZERO RETRANSMIT COUNTER
NO
YES
B
Pge 3
D
Pge 45
SET VCS SWITCH INDICATOR
C
Pge 10

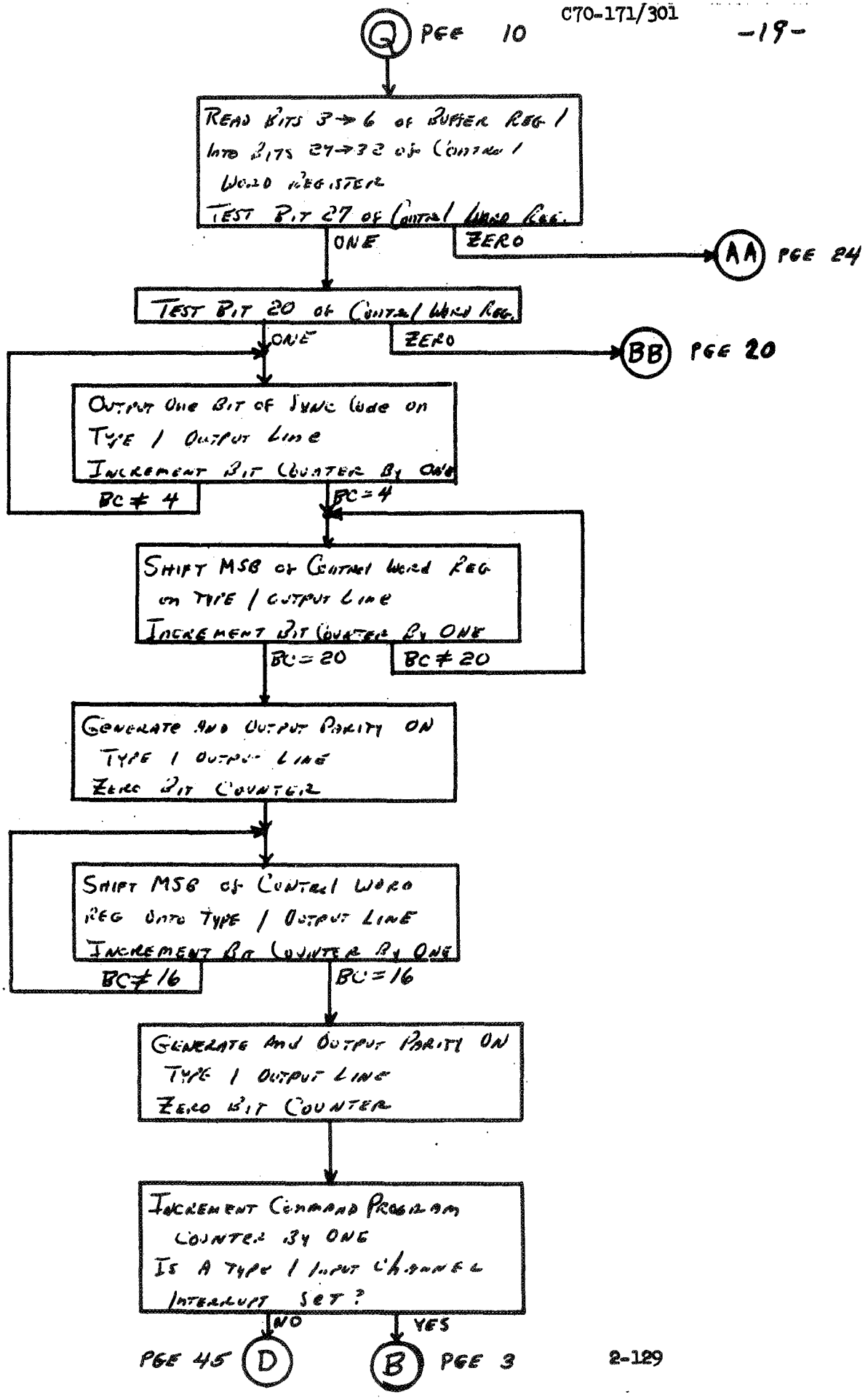

S
PGE 37
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REG.
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB OF BUFFER REG 2 ON OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
SET MEMORY READ LINE FALSE
BC=16
BC≠16
GENERATE AND OUTPUT PARITY AND PLACE IN APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
PGE 16
X
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC≠16
BC=16
U
PGE 15

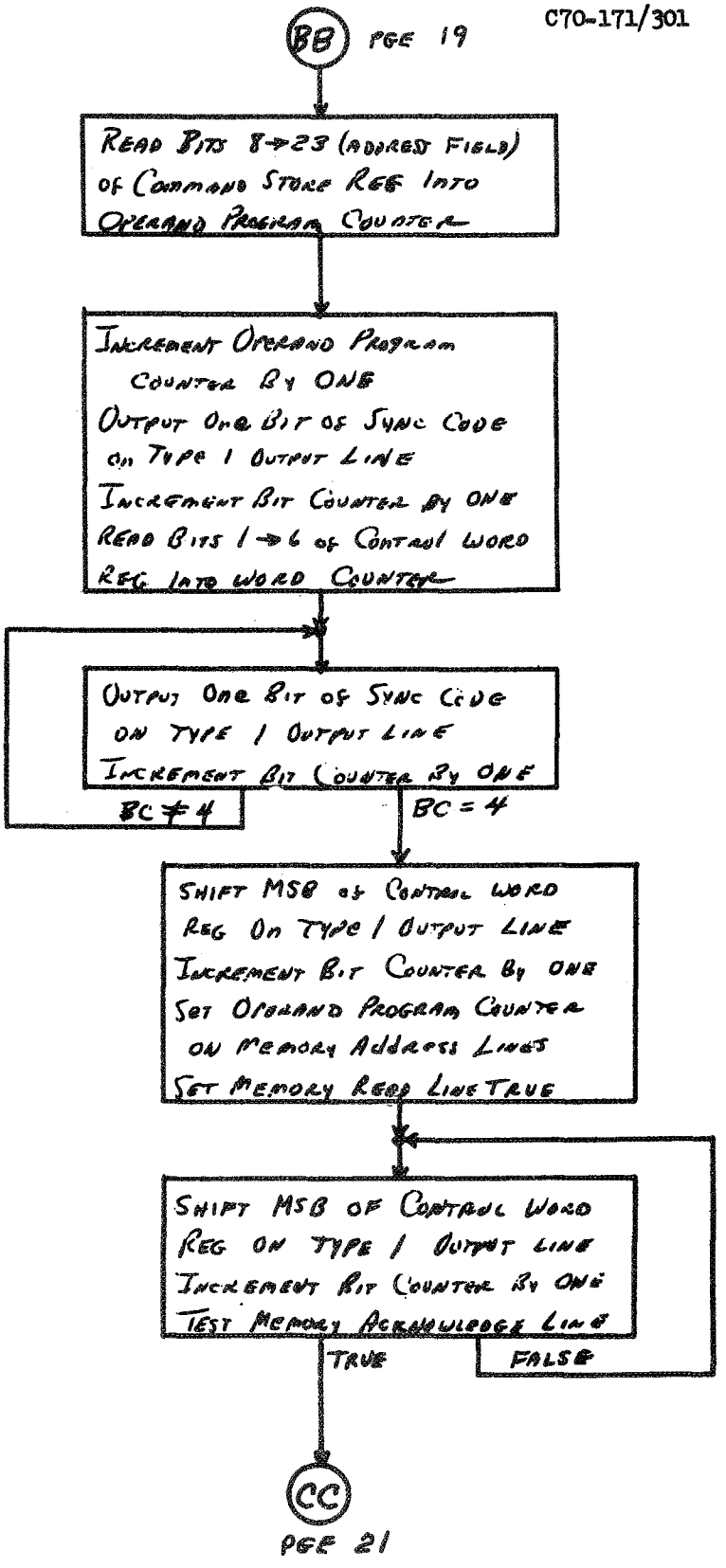

U
PGE 14
GENERATE And OUTPUT PARITY And PLACE IN APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
READ BUFFER REGISTER 1 INTO BUFFER REGISTER 2
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
V
PGE 16
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE And INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER # MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE LINE
FALSE
TRUE
SHIFT MSB OF BUFFER REG 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
W
PGE 16

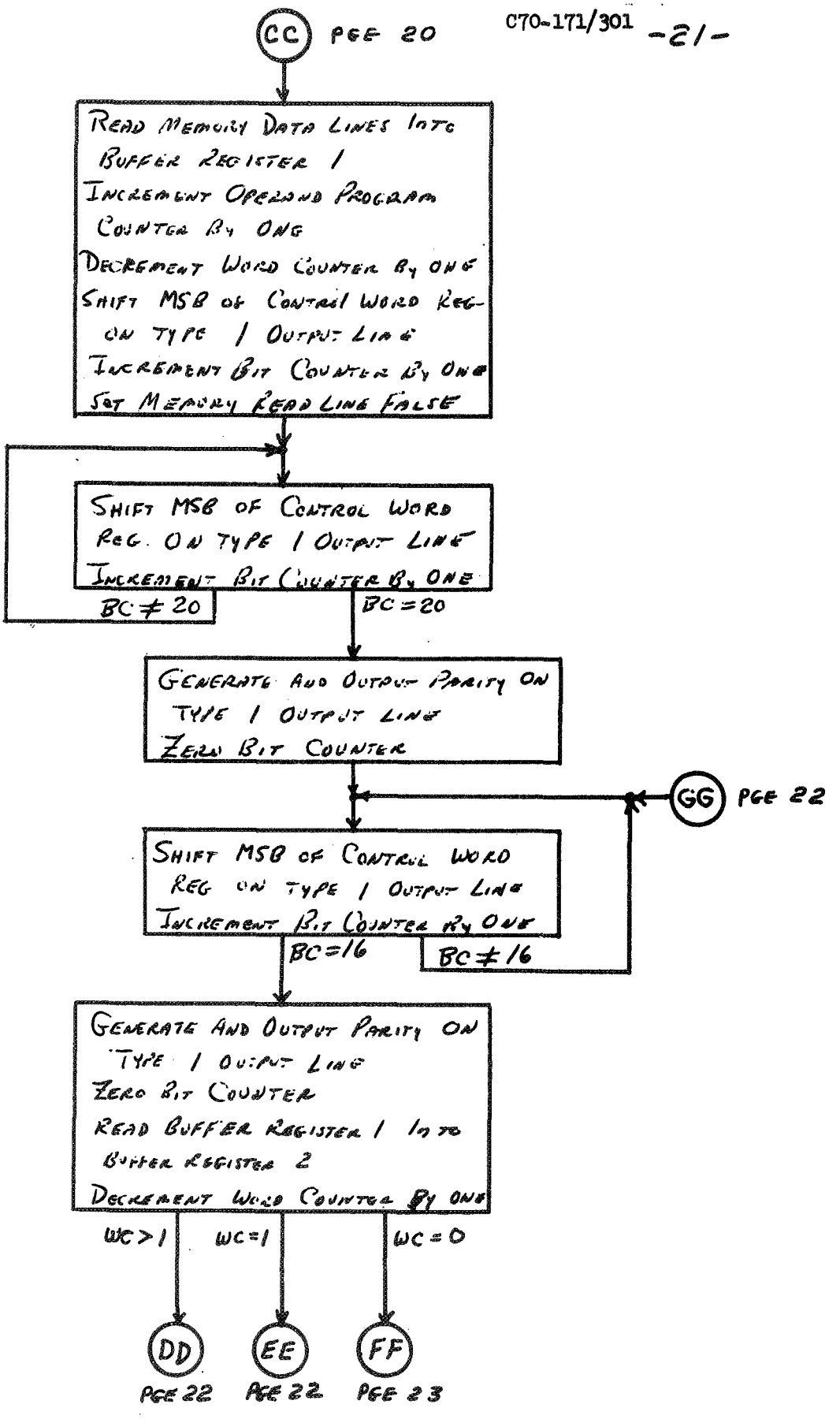

W
PGE 15
SHIFT MSB of BUFFER REG. 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY AND PLACE IN APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
WC = 0
WC ≠ 0
X
PGE 14
V
PGE 15
SHIFT MSB of BUFFER REG 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY AND PLACE INTO APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
ZERO BUFFER REGISTER 2
SHIFT MSB of BUFFER REGISTER 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
INPUT CHANNEL WORD COUNTER ZERO ?
(NO)(BC ≠ 16)
(NO) (BC=16)
(YES) (BC=16)
(YES)(BC ≠ 16)
GENERATE AND OUTPUT PARITY AND PLACE INTO APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
SHIFT MSB of BUFFER REGISTER 2 ONTO OUTPUT LINE and PLACE INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
PGE 17
Y

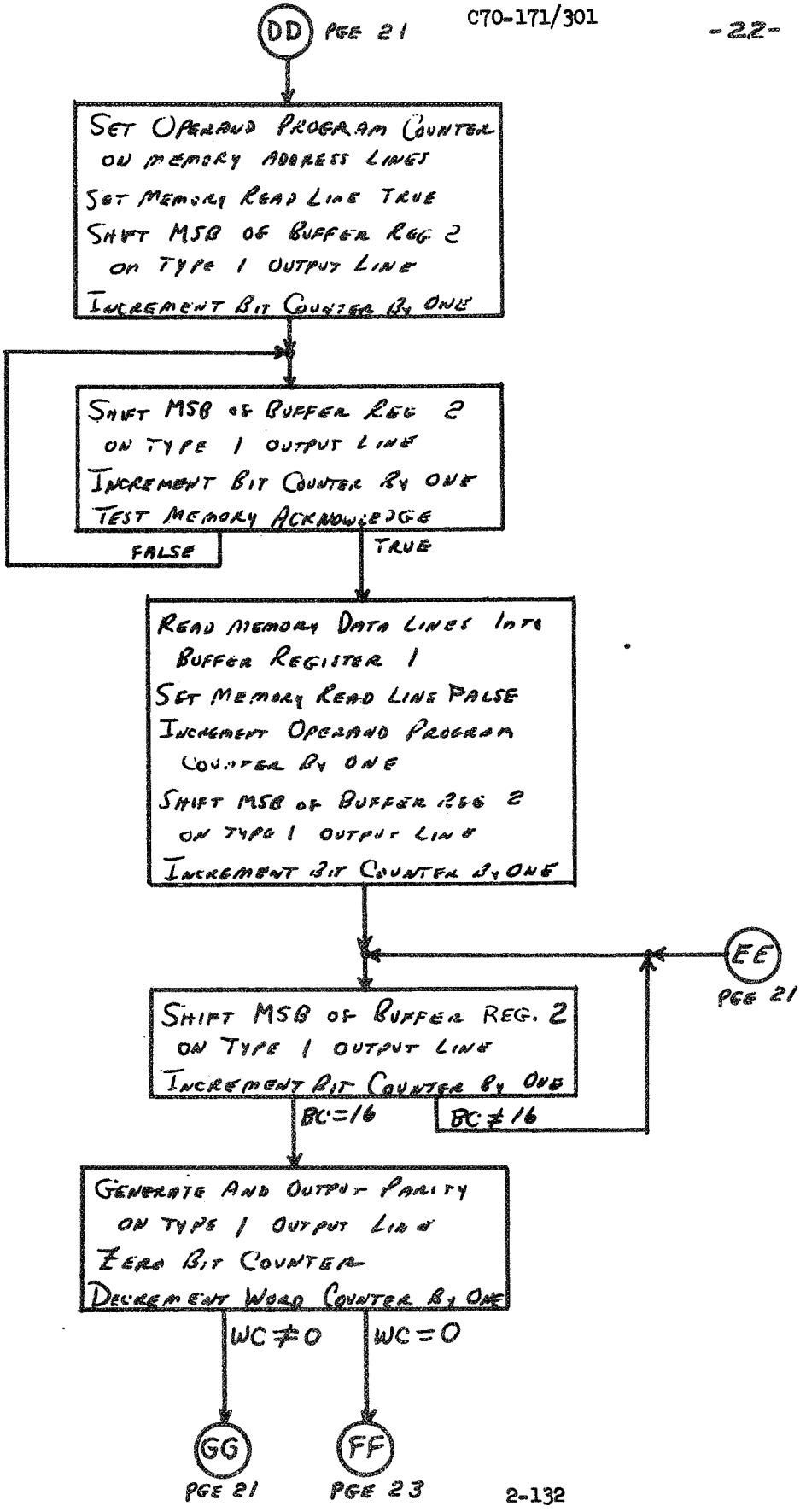

Y
PGE 16
GENERATE AND OUTPUT PARITY AND PLACE INTO APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
READ FIXED CODE INTO BUFFER REG 2
ACCORDING TO COMPARISON RESULTS
SHIFT MSB OF BUFFER REGISTER 2 ONTO OUTPUT LINE AND INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY AND PLACE INTO APPROPRIATE COMPARISON REG.
ZERO BIT COUNTER
IS TYPE 2 INPUT COMPLETE DISCRETE TRUE?
YES
NO
READ BITS 7 → 16 OF BUFFER REG. 2 INTO BITS 1 → 10 OF ACKNOWLEDGE WORD STORE
READ BITS 21 → 26 OF BUFFER REG. 2 INTO BITS 11 → 16 OF ACKNOWLEDGE WORD STORE
J
Pge 18

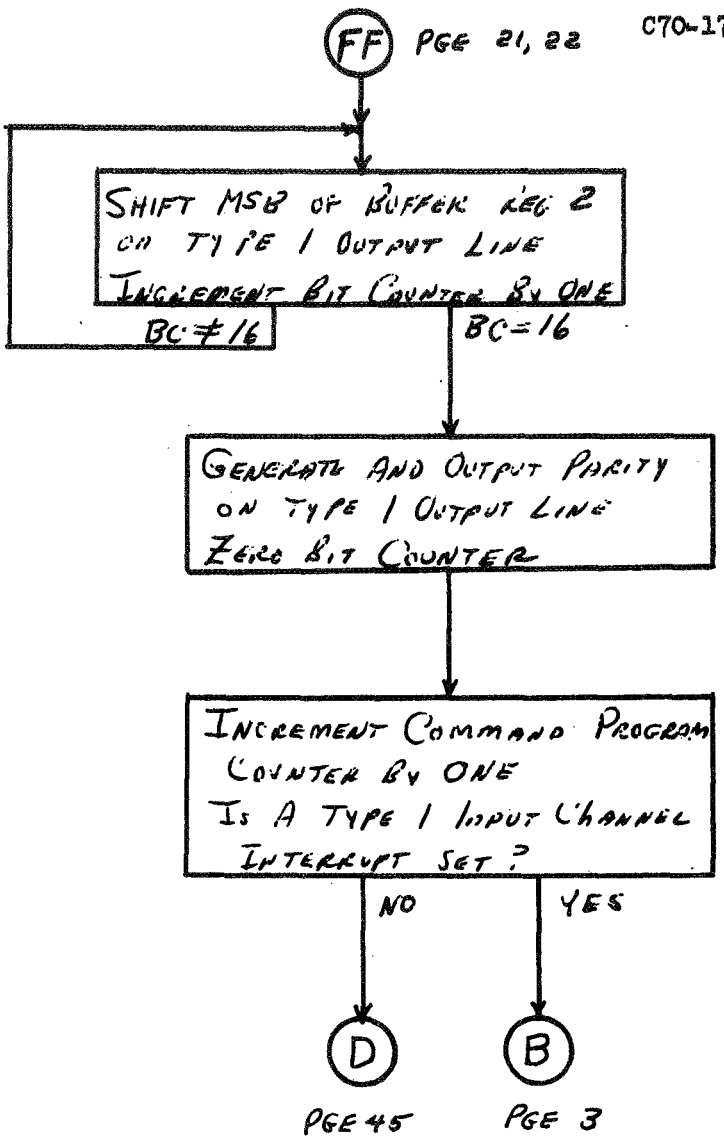

(J) Pge 17
Compare Acknowledge Word Store With Control Word Register:
1. Bits 5→10 with Bits 21→25
2. Bits 11→16 " " 11→16
Set Acknowledge Word Error Indicator And Acknowledge Word Indicator According To Comparison
Form Status Word In Buffer Register 1
Set Operand Program Counter On Memory Address Lines
Set Buffer Register 1 onto Memory Data Lines
Set Memory Write Line True
Test Memory Acknowledge Line
Memory Acknowledge
Set Memory Write Line False
Test Bit 26 of Control Word Register
ONE
ZERO
Was A "Go" Code Received Over Type 1 Input Channel?
NO
YES
Increment Command Program Counter By One
Is A Type 1 Input Channel Interrupt Set?
NO
YES
(D) Pge 45
(B) Pge 3
Increment Retransmit Counter By One
Set Retransmission Indicator
Is Retransmit Counter At 3?
NO
YES
Set VCS Switch Indicator
Zero Retransmit Counter
(C) Pge 10

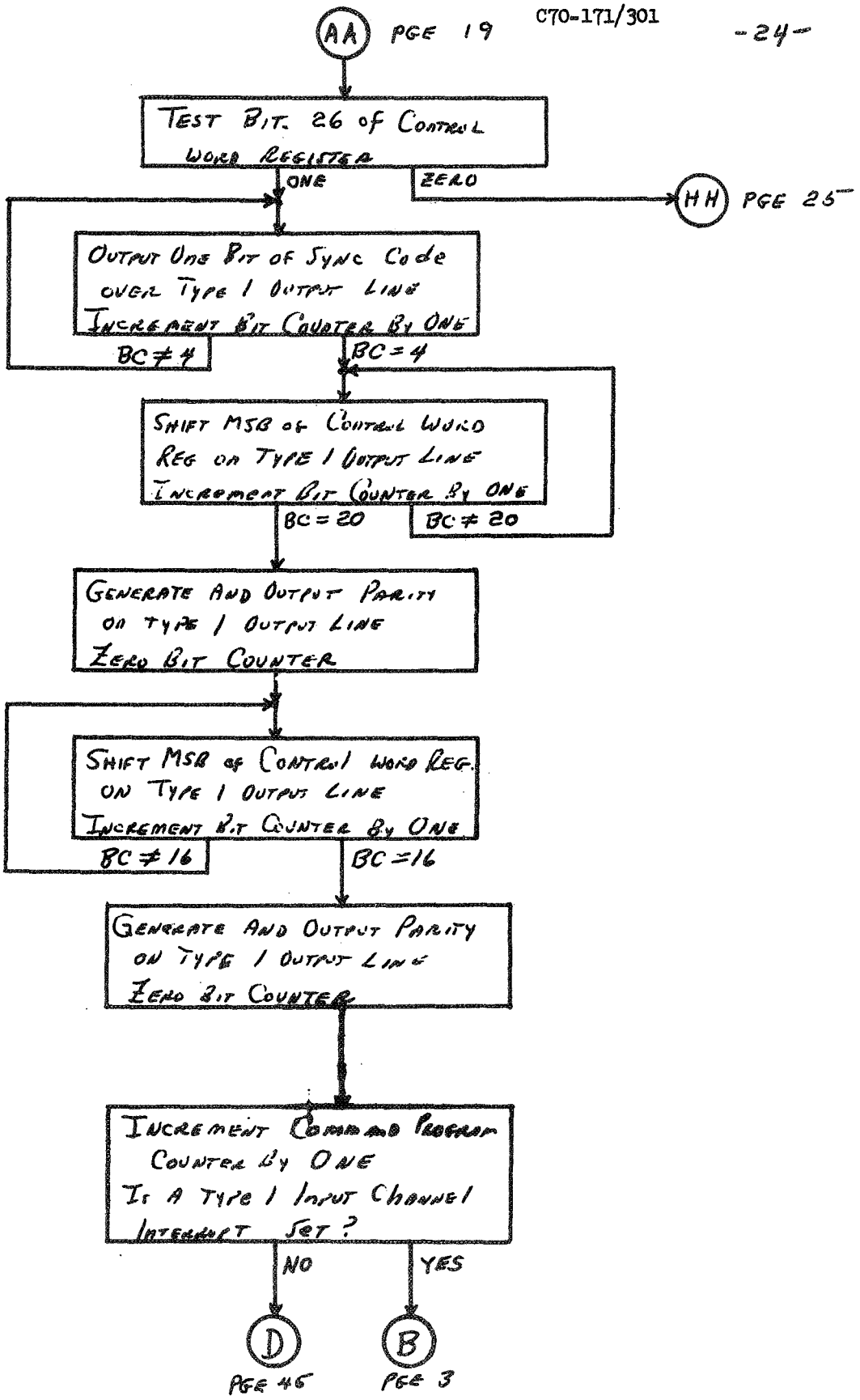

Q PGE 10
READ BITS 3→6 of BUFFER REG / INTO BITS 27→32 of CONTROL / WORD REGISTER
TEST BIT 27 of CONTROL WORD REG.
ONE
ZERO
AA PGE 24
TEST BIT 20 of CONTROL WORD REG.
ONE
ZERO
BB PGE 20
OUTPUT ONE BIT OF SYNC CODE ON TYPE / OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 4
BC = 4
SHIFT MSB OF CONTROL WORD REG ON TYPE / OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 20
BC ≠ 20
GENERATE AND OUTPUT PARITY ON TYPE / OUTPUT LINE
ZERO BIT COUNTER
SHIFT MSB OF CONTROL WORD REG INTO TYPE / OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY ON TYPE / OUTPUT LINE
ZERO BIT COUNTER
INCREMENT COMMAND PROGRAM COUNTER BY ONE
IS A TYPE / INPUT CHANNEL INTERRUPT SET?
NO
YES
PGE 45 D
B PGE 3

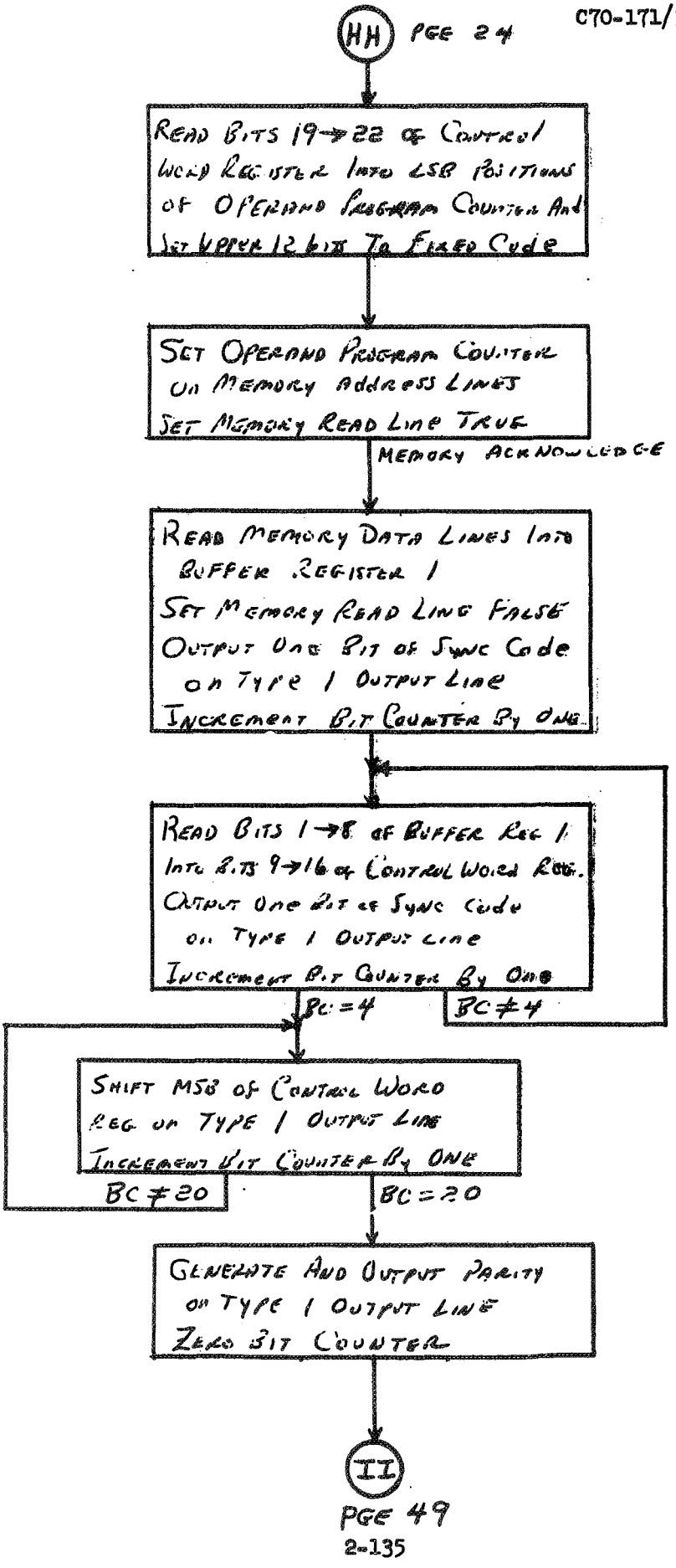

BB
PGE 19
READ BITS 8→23 (ADDRESS FIELD) OF COMMAND STORE REG INTO OPERAND PROGRAM COUNTER
INCREMENT OPERAND PROGRAM COUNTER BY ONE
OUTPUT ONE BIT OF SYNC CODE ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ BITS 1→6 OF CONTROL WORD REG INTO WORD COUNTER
OUTPUT ONE BIT OF SYNC CODE ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 4
BC = 4
SHIFT MSB OF CONTROL WORD REG ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB OF CONTROL WORD REG ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE LINE
TRUE
FALSE
CC
PGE 21

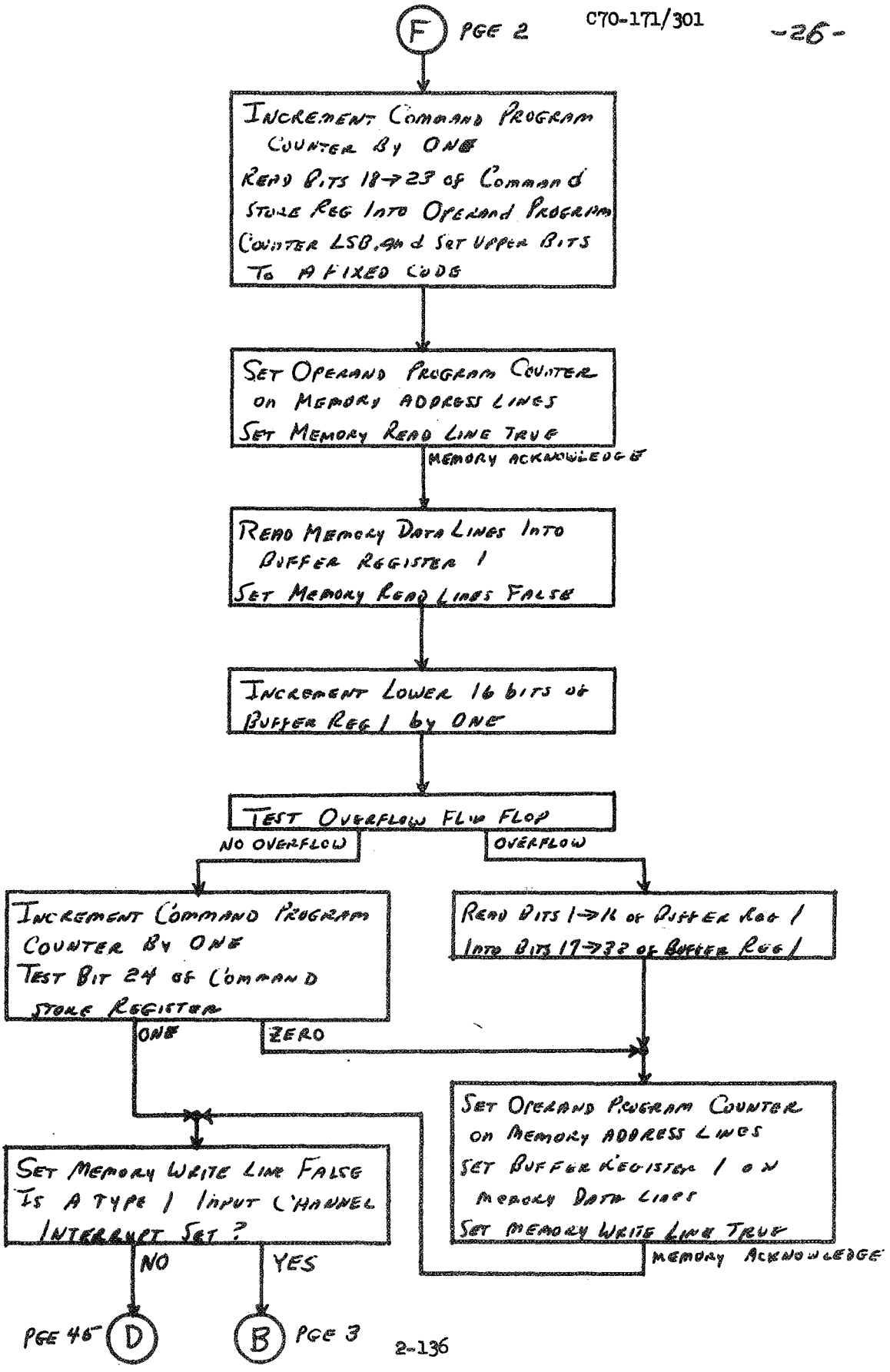

CC
PGE 20
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
INCREMENT OPERAND PROGRAM COUNTER BY ONE
DECREMENT WORD COUNTER BY ONE
SHIFT MSB OF CONTROL WORD REG ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SET MEMORY READ LINE FALSE
SHIFT MSB OF CONTROL WORD REG. ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 20
BC = 20
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
GG
PGE 22
SHIFT MSB OF CONTROL WORD REG ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
READ BUFFER REGISTER 1 INTO BUFFER REGISTER 2
DECREMENT WORD COUNTER BY ONE
WC > 1
WC = 1
WC = 0
DD
PGE 22
EE
PGE 22
FF
PGE 23

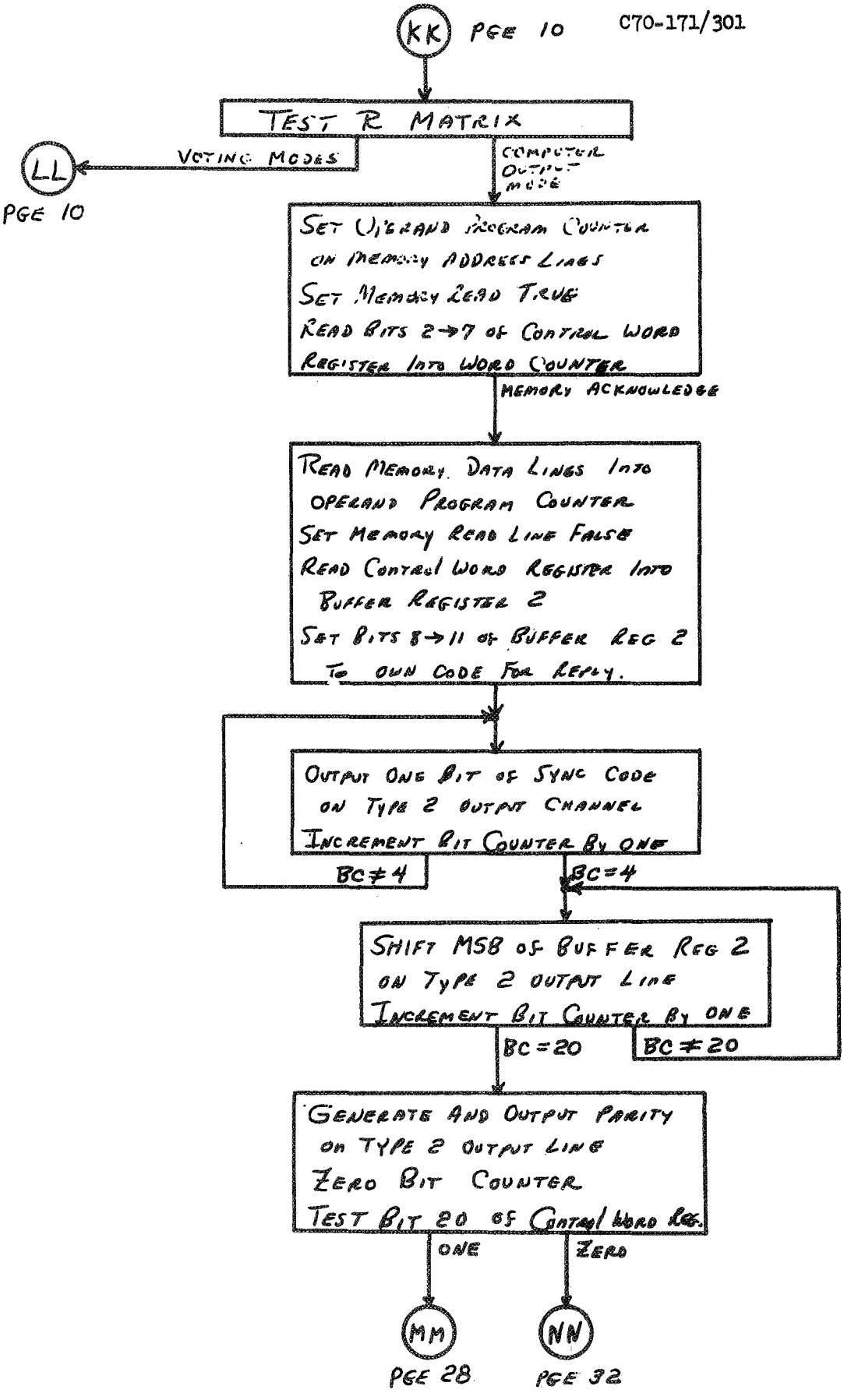

DD
PGE 21
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB OF BUFFER REG 2 ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SHIFT MSB OF BUFFER REG 2 ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SHIFT MSB OF BUFFER REG 2 ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
EE
PGE 21
SHIFT MSB OF BUFFER REG. 2 ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=16
BC≠16
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
WC≠0
WC=0
GG
PGE 21
FF
PGE 23

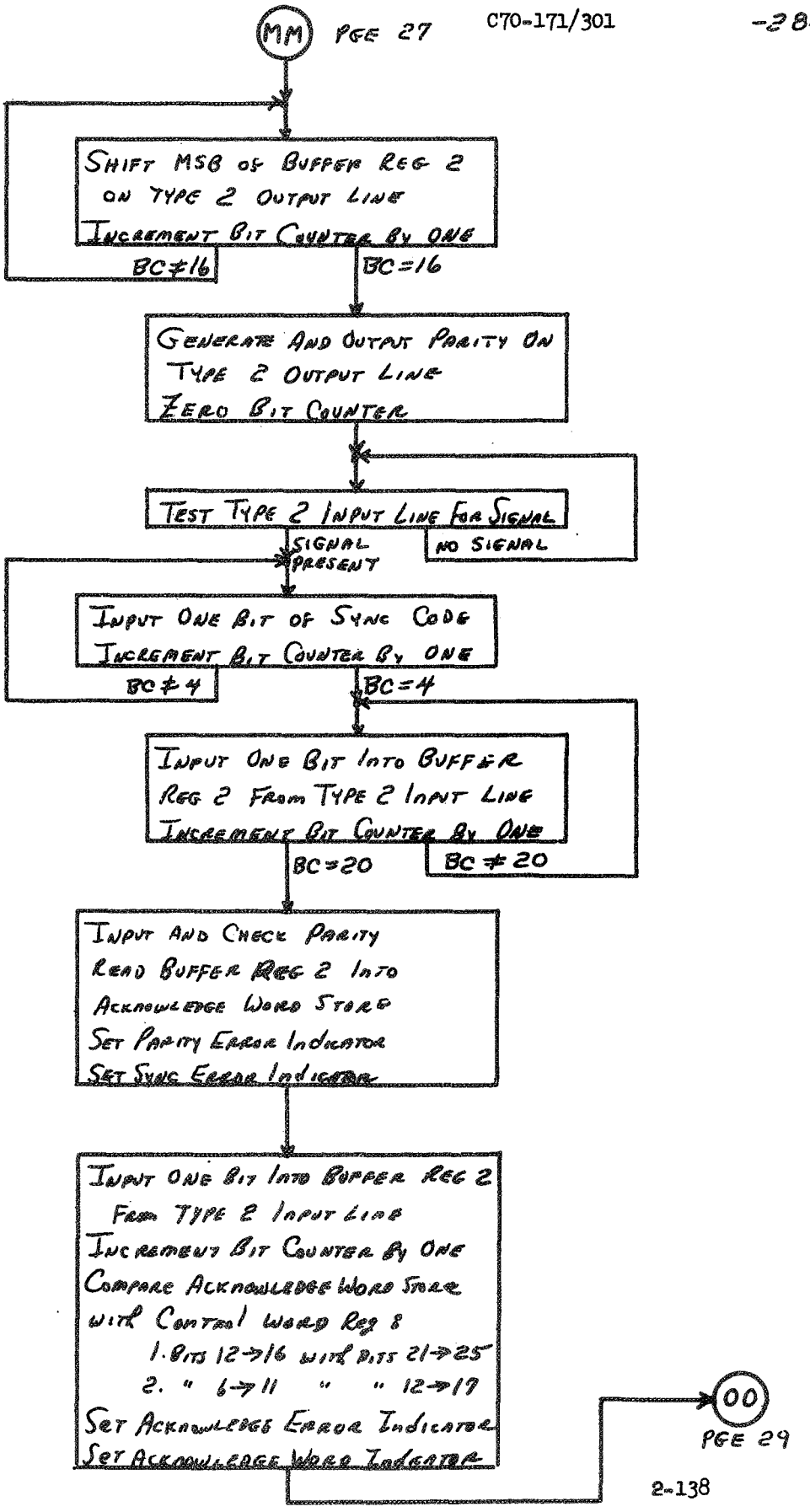

FF
PGE 21, 22
SHIFT MSB OF BUFFER REG 2 ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC≠16
BC=16
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
INCREMENT COMMAND PROGRAM COUNTER BY ONE
IS A TYPE 1 INPUT CHANNEL INTERRUPT SET?
NO
YES
D
PGE 45
B
PGE 3

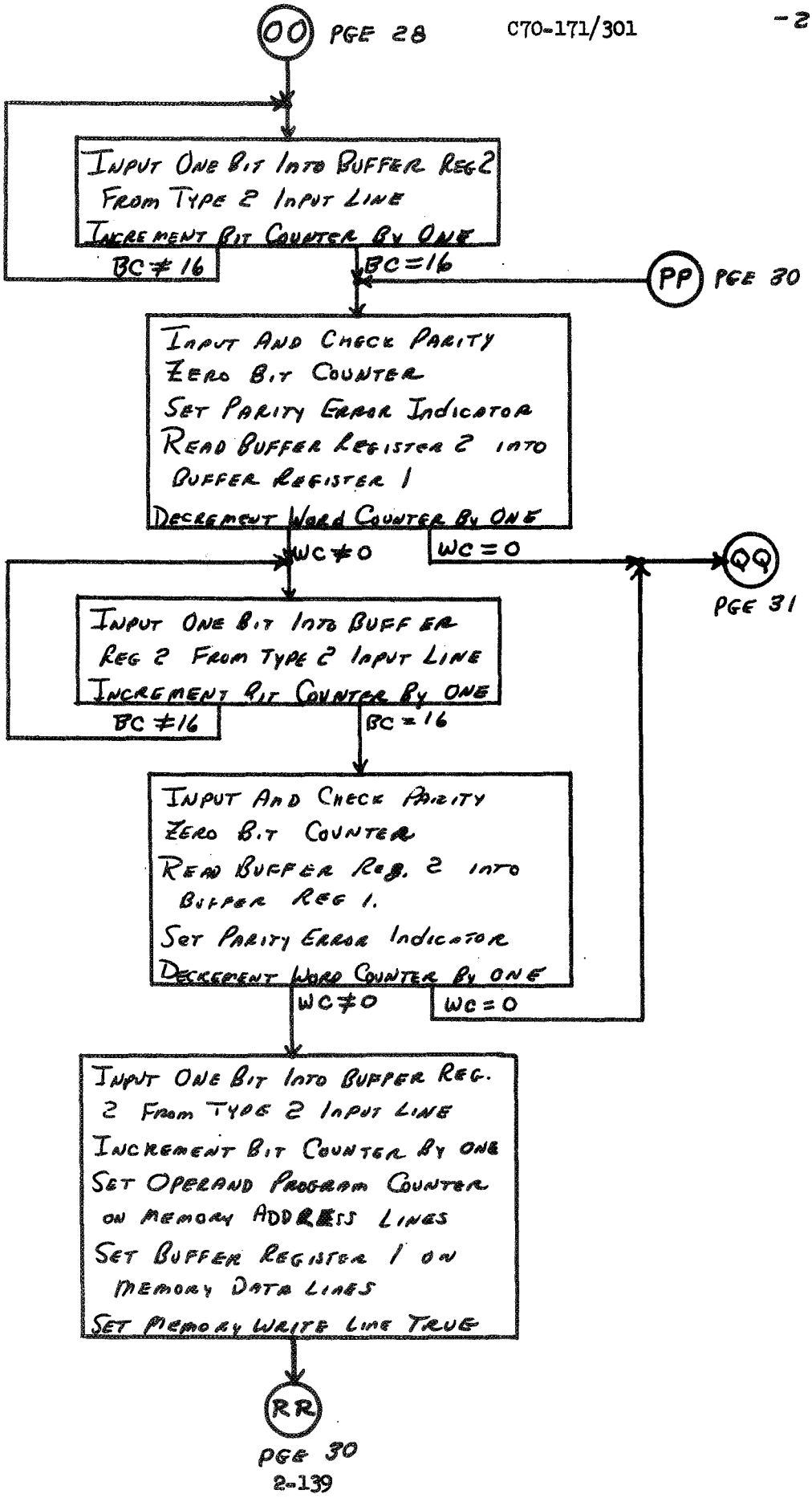

AA
PGE 19
TEST BIT 26 of CONTROL WORD REGISTER
ONE
ZERO
HH
PGE 25
OUTPUT ONE BIT OF SYNC CODE OVER TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 4
BC = 4
SHIFT MSB of CONTROL WORD REG ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 20
BC ≠ 20
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
SHIFT MSB of CONTROL WORD REG. ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
INCREMENT COMMAND PROGRAM COUNTER BY ONE
IS A TYPE 1 INPUT CHANNEL INTERRUPT SET?
NO
YES
D
PGE 46
B
PGE 3

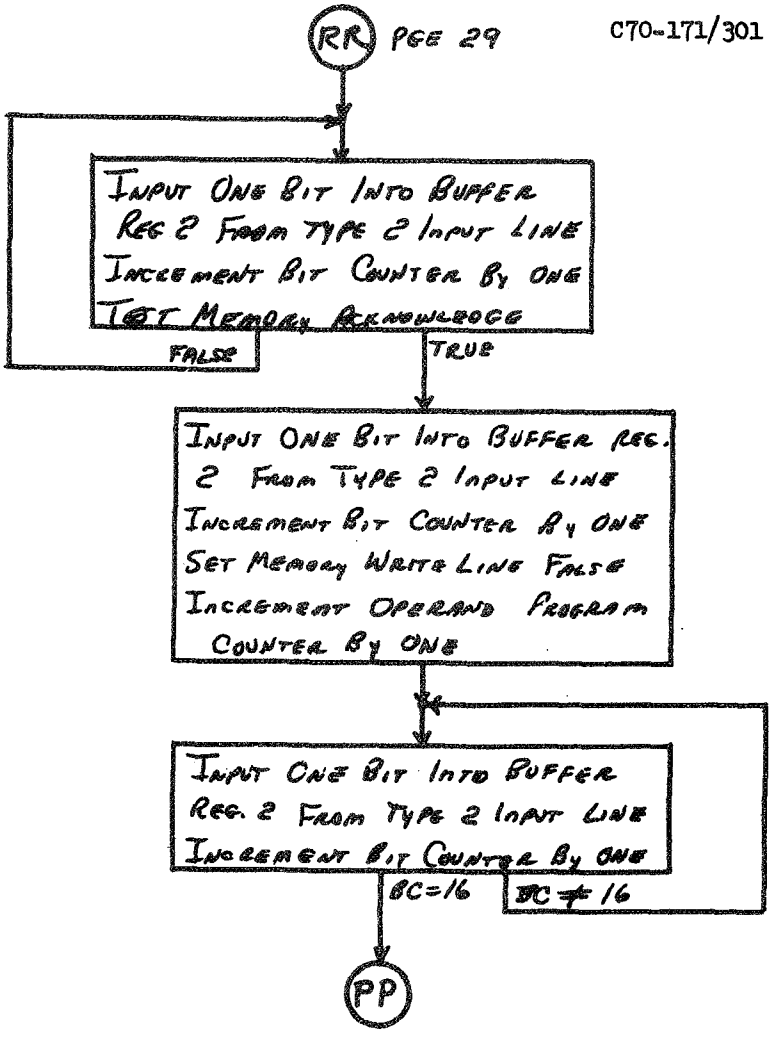

HH
PGE 24
READ BITS 19→22 OF CONTROL WORD REGISTER INTO LSB POSITIONS OF OPERAND PROGRAM COUNTER AND SET UPPER 12 BITS TO FIXED CODE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
OUTPUT ONE BIT OF SYNC CODE ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ BITS 1→8 OF BUFFER REG 1 INTO BITS 9→16 OF CONTROL WORD REG.
OUTPUT ONE BIT OF SYNC CODE ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=4
BC≠4
SHIFT MSB OF CONTROL WORD REG ON TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC≠20
BC=20
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
ZERO BIT COUNTER
II
PGE 49

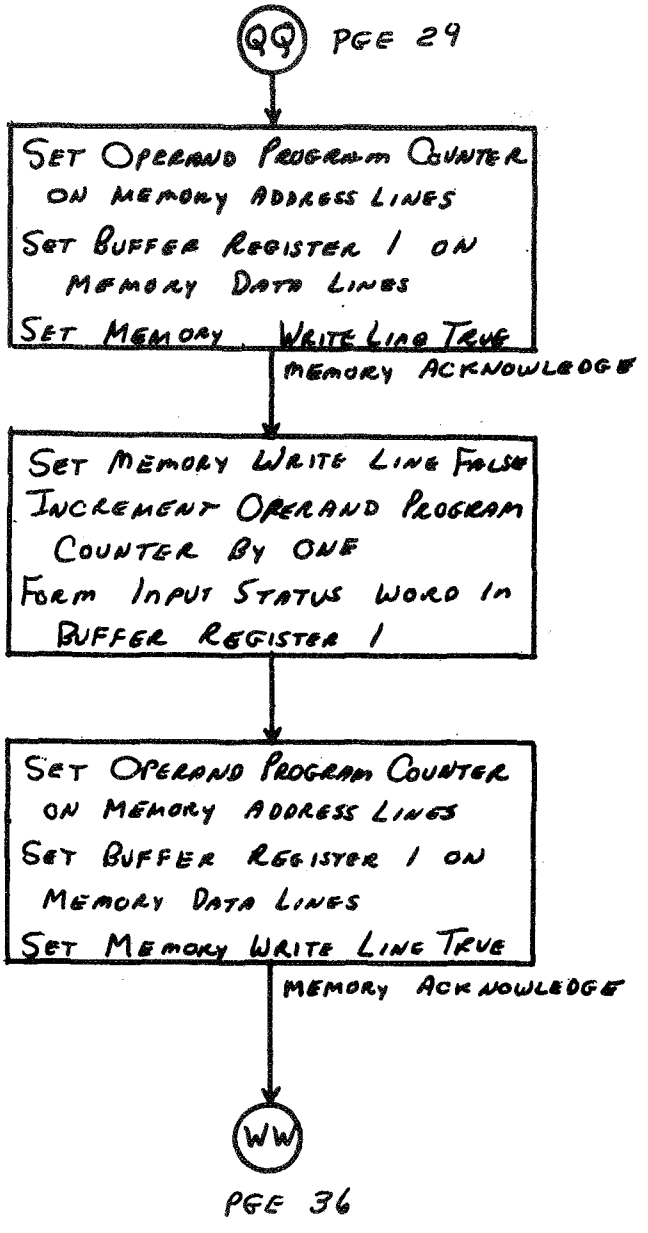

F PGE 2
INCREMENT COMMAND PROGRAM COUNTER BY ONE
READ BITS 18→23 OF COMMAND STORE REG INTO OPERAND PROGRAM COUNTER LSB, AND SET UPPER BITS TO A FIXED CODE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINES FALSE
INCREMENT LOWER 16 BITS OF BUFFER REG 1 BY ONE
TEST OVERFLOW FLIP FLOP
NO OVERFLOW
OVERFLOW
INCREMENT COMMAND PROGRAM COUNTER BY ONE
TEST BIT 24 OF COMMAND STORE REGISTER
READ BITS 1→16 OF BUFFER REG 1 INTO BITS 17→32 OF BUFFER REG 1
ONE
ZERO
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
SET MEMORY WRITE LINE FALSE
IS A TYPE 1 INPUT CHANNEL INTERRUPT SET?
NO
YES
PGE 46 D
B PGE 3

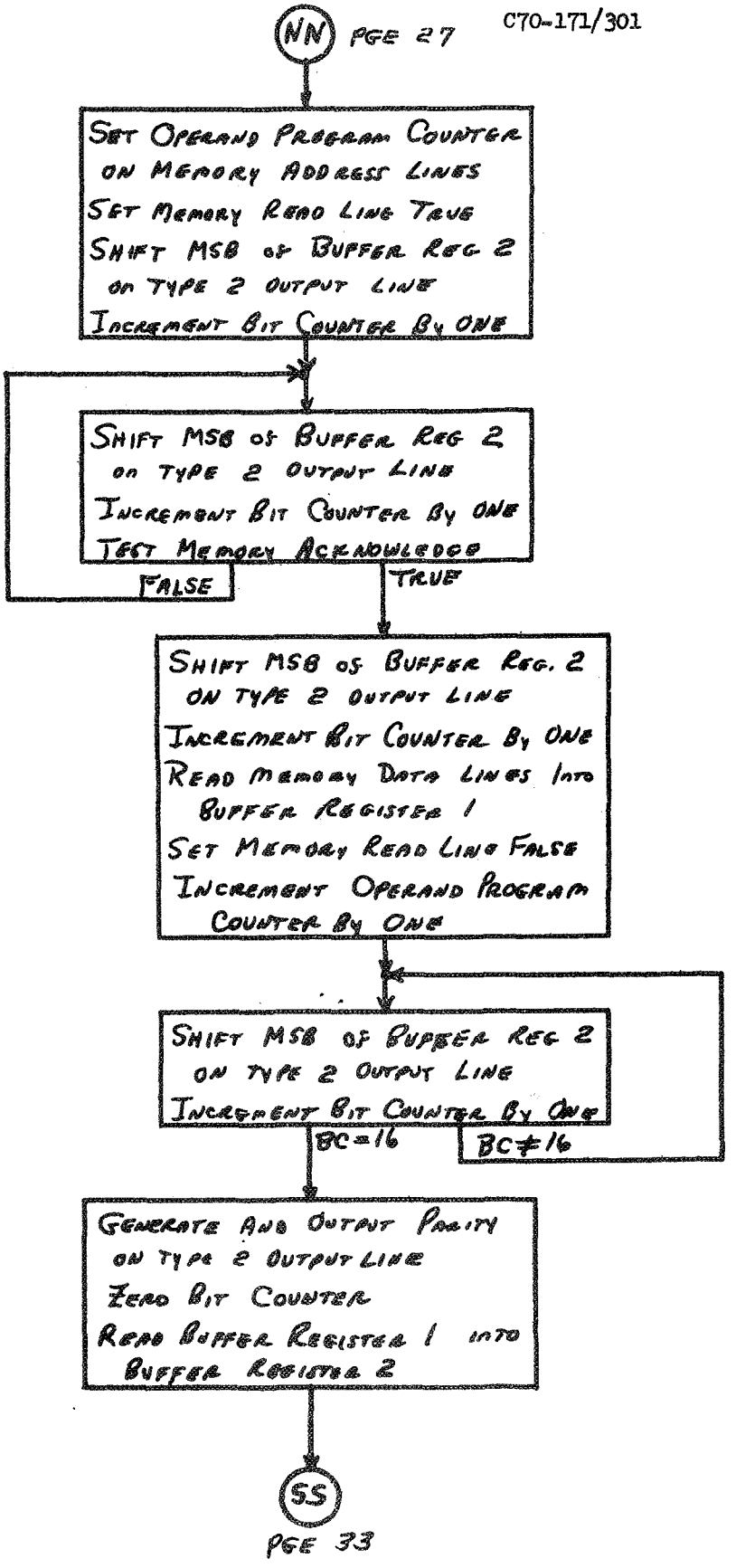

KK
PGE 10
TEST R MATRIX
VOTING MODES
COMPUTER OUTPUT MODE
LL
PGE 10
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ TRUE
READ BITS 2→7 OF CONTROL WORD REGISTER INTO WORD COUNTER
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO OPERAND PROGRAM COUNTER
SET MEMORY READ LINE FALSE
READ CONTROL WORD REGISTER INTO BUFFER REGISTER 2
SET BITS 8→11 OF BUFFER REG 2 TO OWN CODE FOR REPLY.
OUTPUT ONE BIT OF SYNC CODE ON TYPE 2 OUTPUT CHANNEL
INCREMENT BIT COUNTER BY ONE
BC≠4
BC=4
SHIFT MSB OF BUFFER REG 2 ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=20
BC≠20
GENERATE AND OUTPUT PARITY ON TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
TEST BIT 20 OF CONTROL WORD REG.
ONE
ZERO
MM
PGE 28
NN
PGE 32

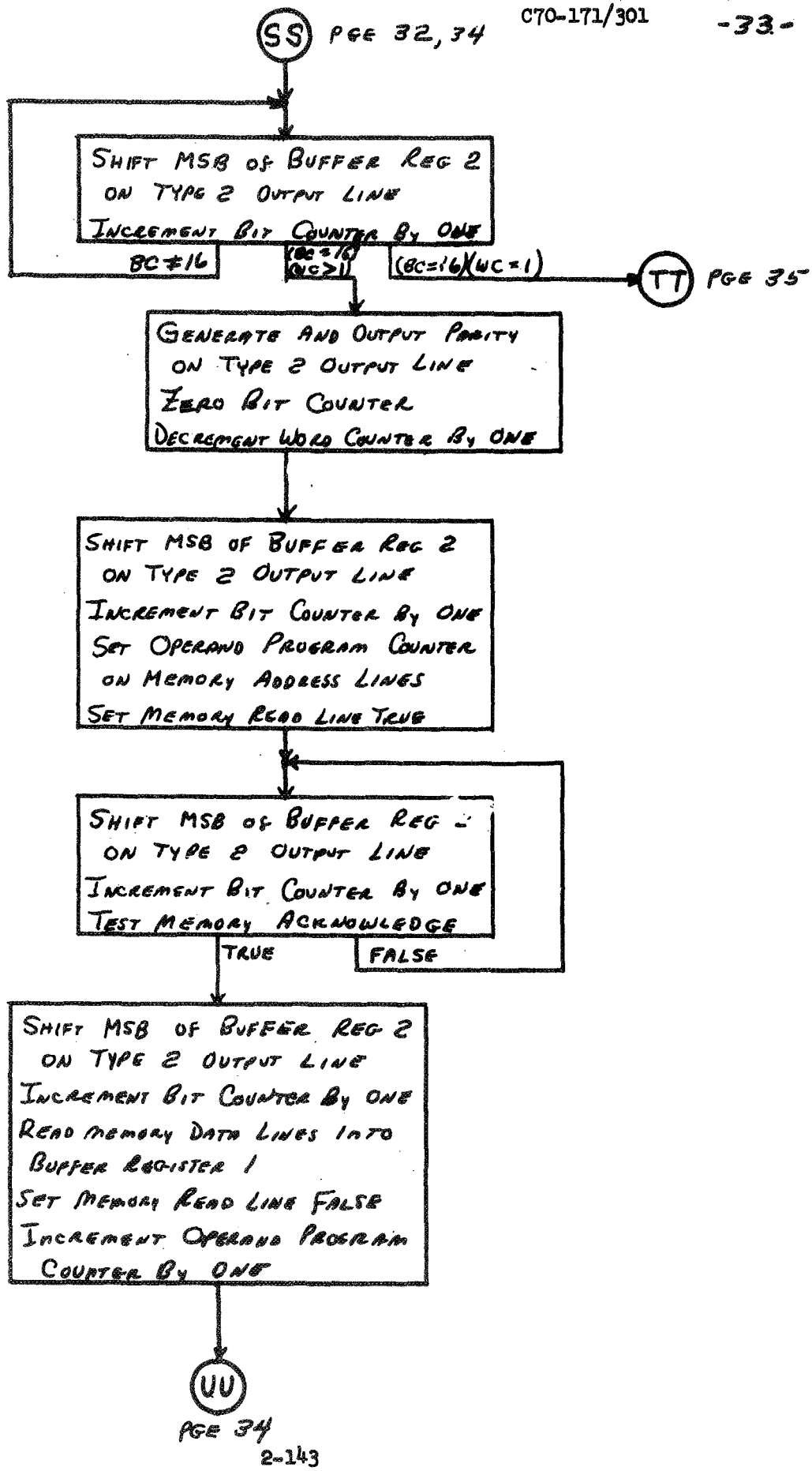

MM
PGE 27
SHIFT MSB OF BUFFER REG 2
ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC≠16
BC=16
GENERATE AND OUTPUT PARITY ON
TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
TEST TYPE 2 INPUT LINE FOR SIGNAL
SIGNAL PRESENT
NO SIGNAL
INPUT ONE BIT OF SYNC CODE
INCREMENT BIT COUNTER BY ONE
BC≠4
BC=4
INPUT ONE BIT INTO BUFFER
REG 2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=20
BC≠20
INPUT AND CHECK PARITY
READ BUFFER REG 2 INTO
ACKNOWLEDGE WORD STORE
SET PARITY ERROR INDICATOR
SET SYNC ERROR INDICATOR
INPUT ONE BIT INTO BUFFER REG 2
FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
COMPARE ACKNOWLEDGE WORD STORE
WITH CONTROL WORD REG 8
1. BITS 12→16 WITH BITS 21→25
2. " 6→11 " " 12→19
SET ACKNOWLEDGE ERROR INDICATOR
SET ACKNOWLEDGE WORD INDICATOR
00
PGE 29

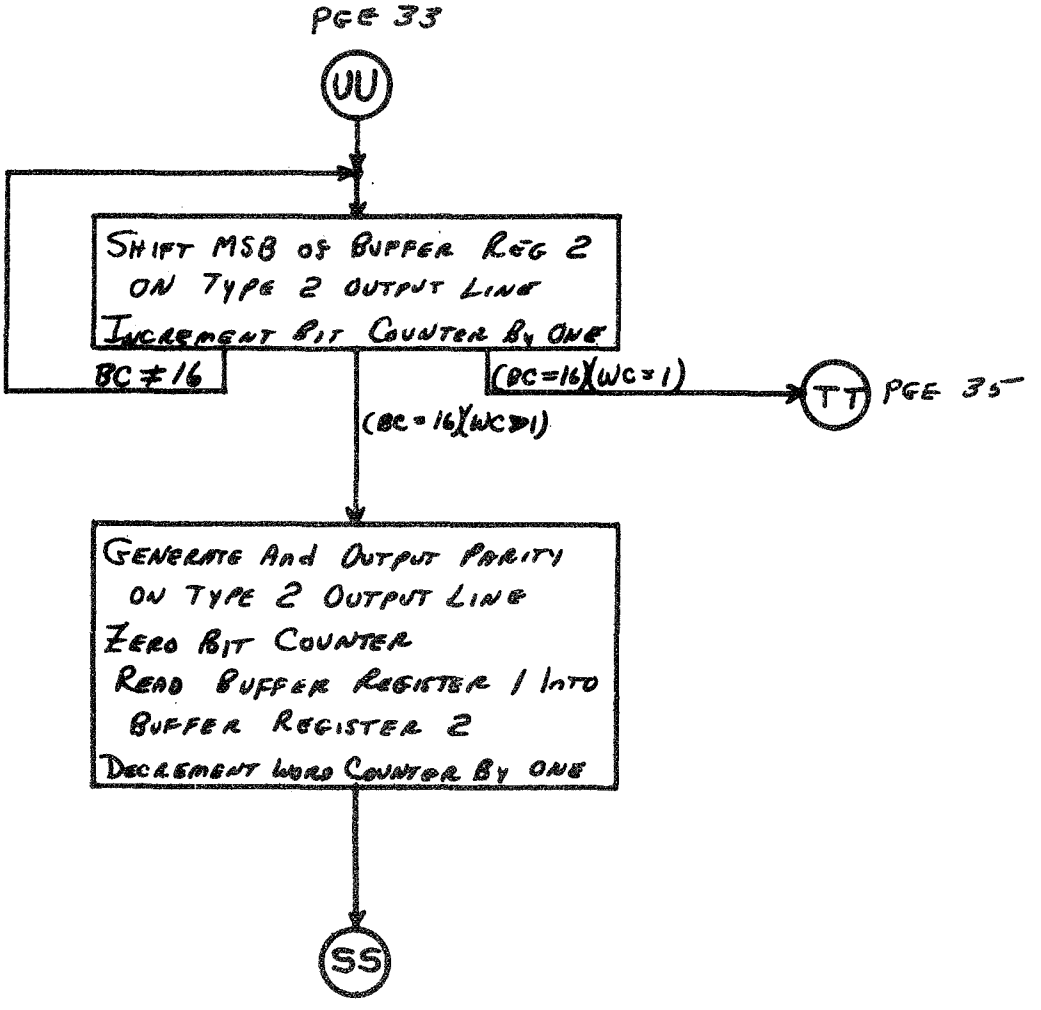

OO
PGE 28
INPUT ONE BIT INTO BUFFER REG 2
FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
PP
PGE 30
INPUT AND CHECK PARITY
ZERO BIT COUNTER
SET PARITY ERROR INDICATOR
READ BUFFER REGISTER 2 INTO
BUFFER REGISTER 1
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
QQ
PGE 31
INPUT ONE BIT INTO BUFFER
REG 2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
INPUT AND CHECK PARITY
ZERO BIT COUNTER
READ BUFFER REG. 2 INTO
BUFFER REG 1.
SET PARITY ERROR INDICATOR
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
INPUT ONE BIT INTO BUFFER REG.
2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER
ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON
MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
RR
PGE 30

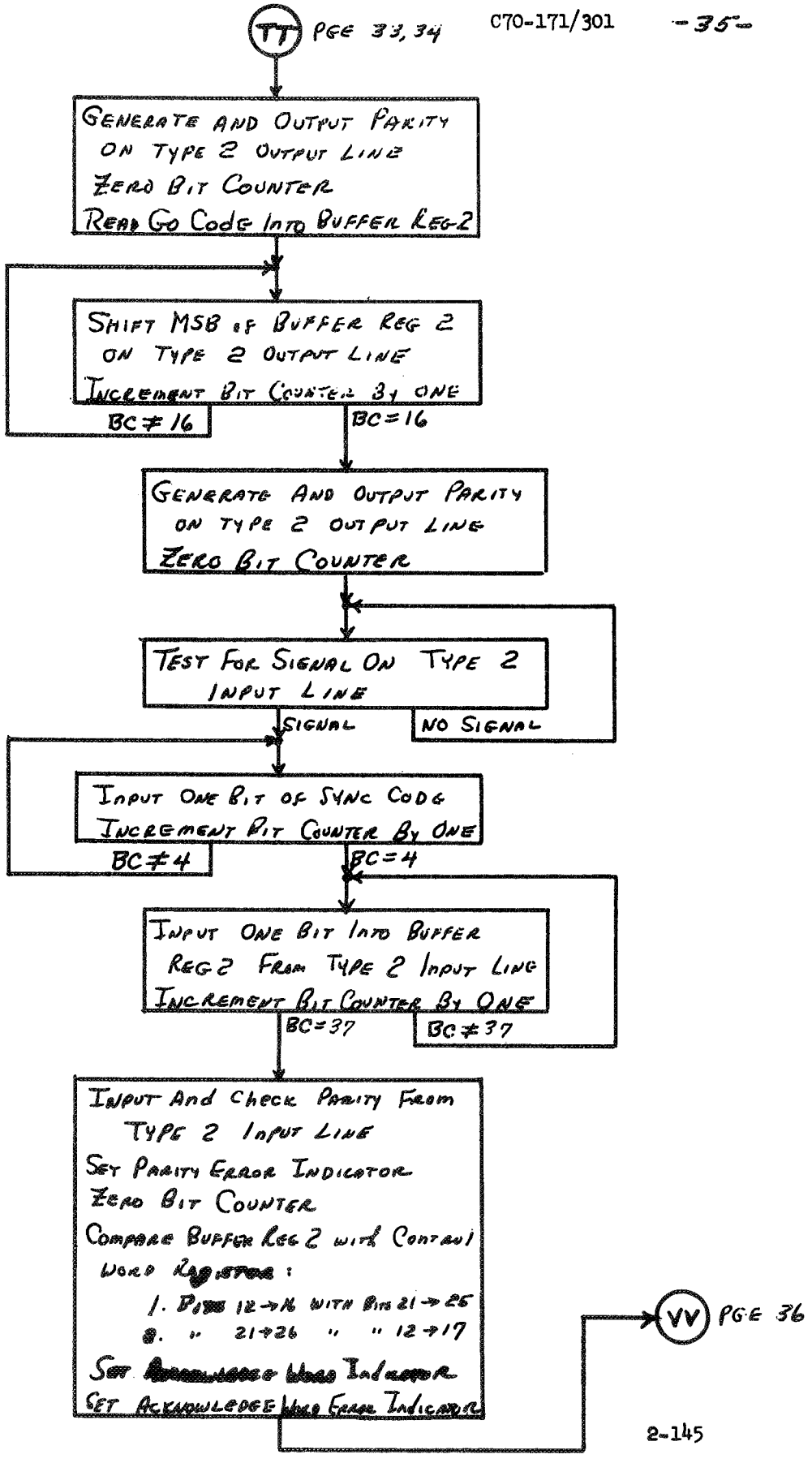

RR
PGE 29
INPUT ONE BIT INTO BUFFER REG 2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
INPUT ONE BIT INTO BUFFER REG. 2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
SET MEMORY WRITE LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
INPUT ONE BIT INTO BUFFER REG. 2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=16
BC≠16
PP
PGE 29

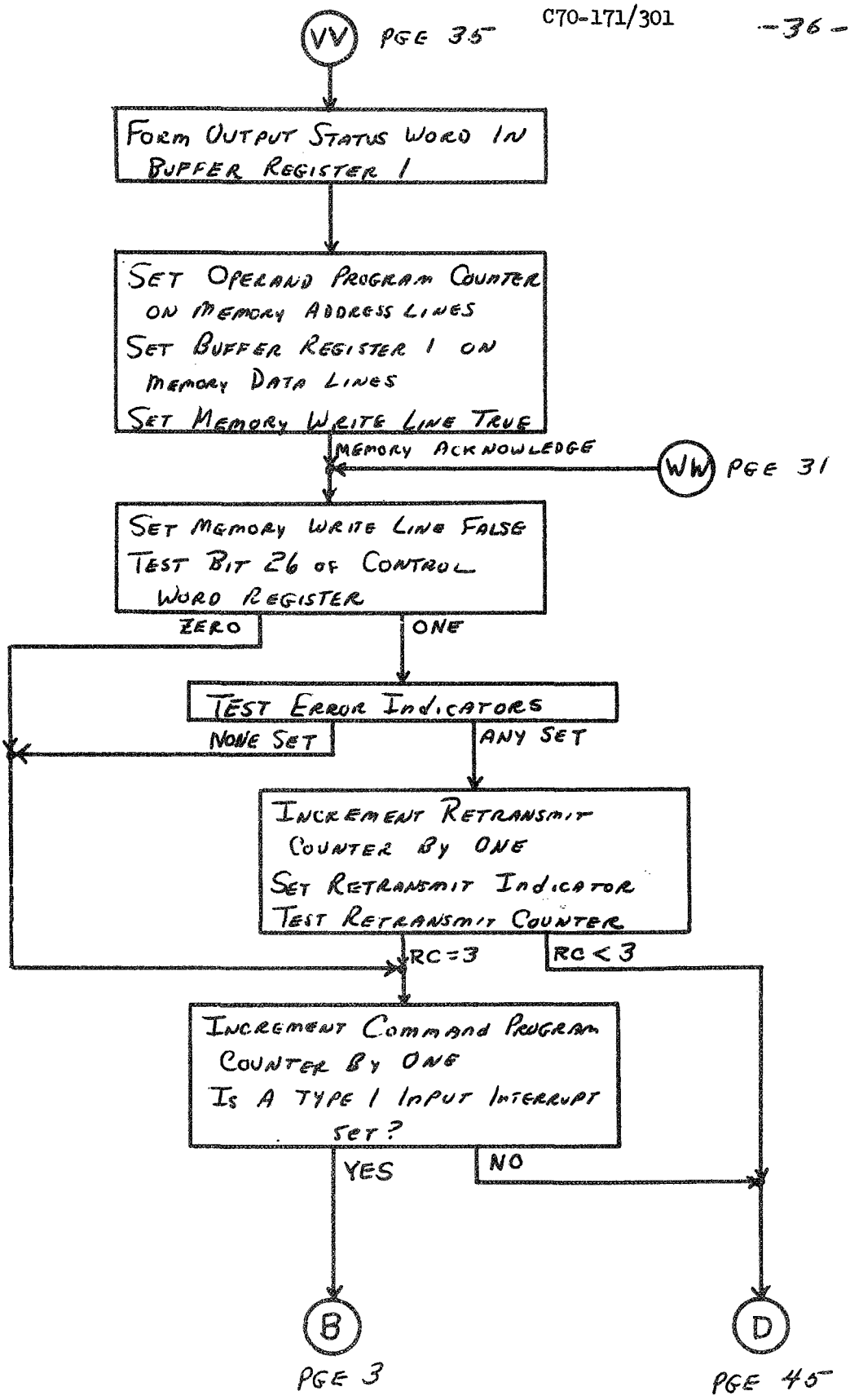

QQ
PGE 29
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
SET MEMORY WRITE LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
FORM INPUT STATUS WORD IN BUFFER REGISTER 1
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
WW
PGE 36

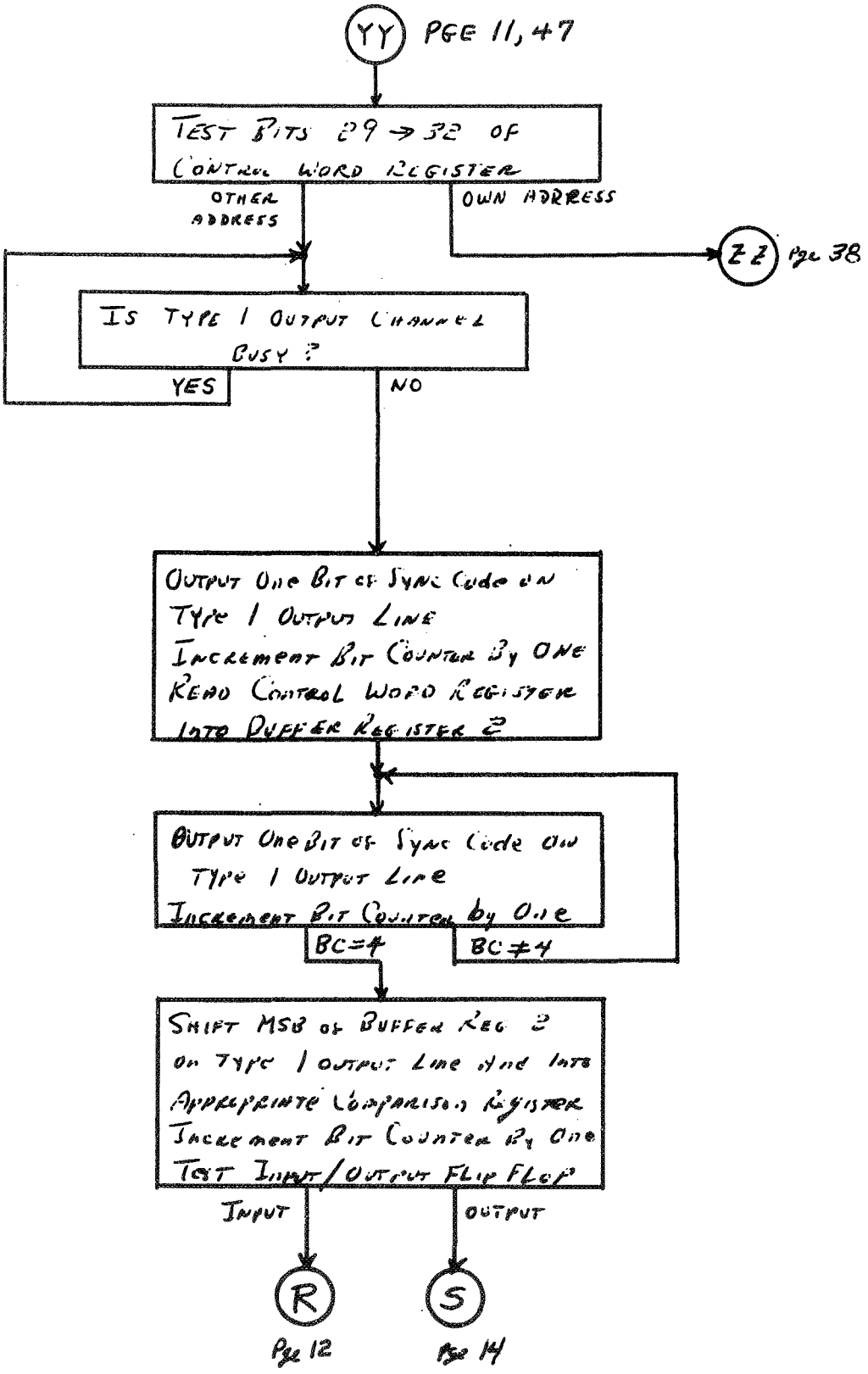

NN
PGE 27
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB of BUFFER REG 2 ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SHIFT MSB of BUFFER REG 2 ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
FALSE
TRUE
SHIFT MSB of BUFFER REG. 2 ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SHIFT MSB of BUFFER REG 2 ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC=16
BC≠16
GENERATE AND OUTPUT PARITY ON TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
READ BUFFER REGISTER 1 INTO BUFFER REGISTER 2
SS
PGE 33

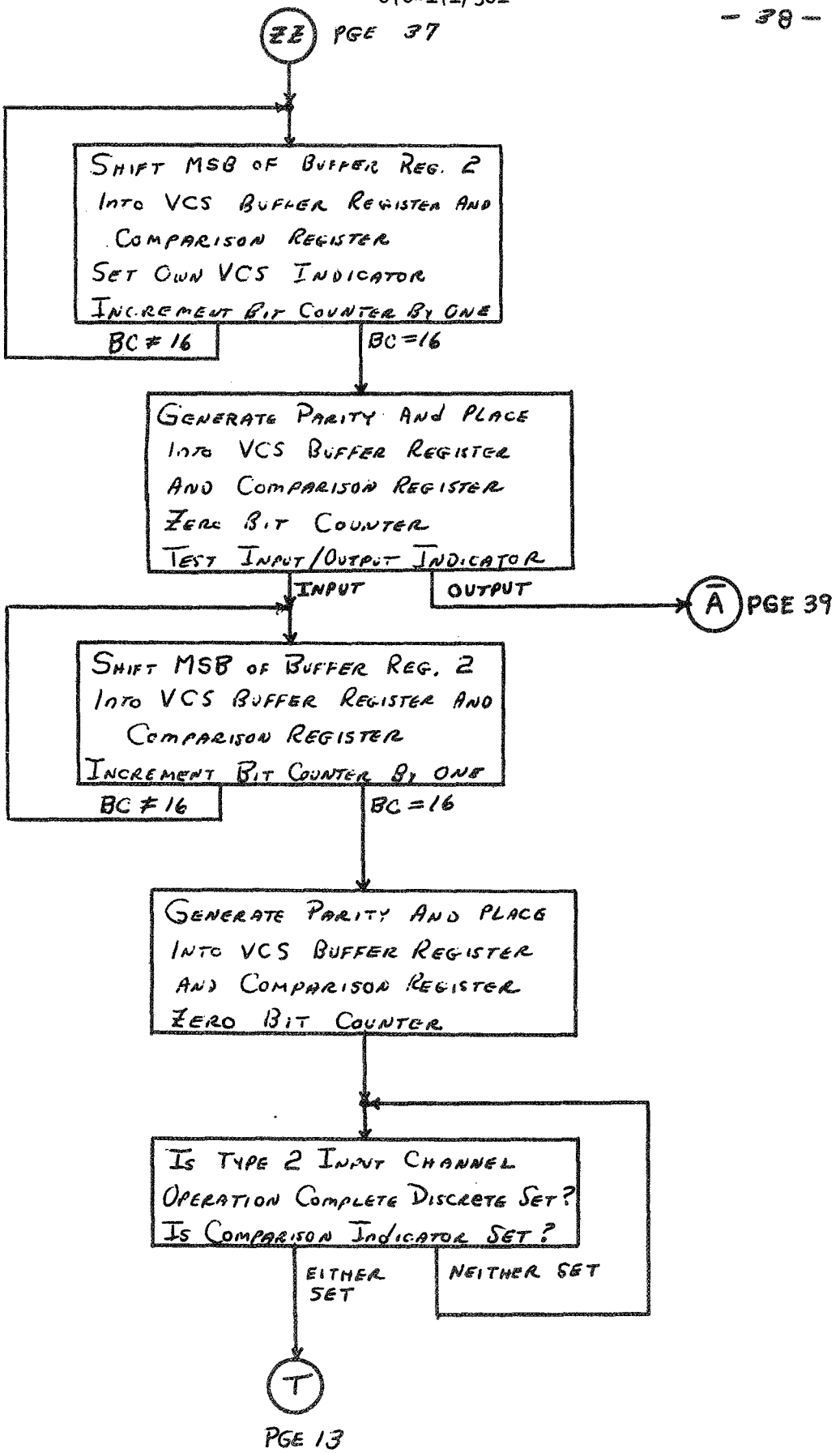

SS
PGE 32, 34
SHIFT MSB of BUFFER REG 2
ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC≠16
(BC=16)(WC>1)
(BC=16)(WC=1)
TT
PGE 35
GENERATE AND OUTPUT PARITY
ON TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
DECREMENT WORD COUNTER BY ONE
SHIFT MSB OF BUFFER REG 2
ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER
ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB OF BUFFER REG 2
ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE
TRUE
FALSE
SHIFT MSB OF BUFFER REG 2
ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO
BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM
COUNTER BY ONE
UU
PGE 34

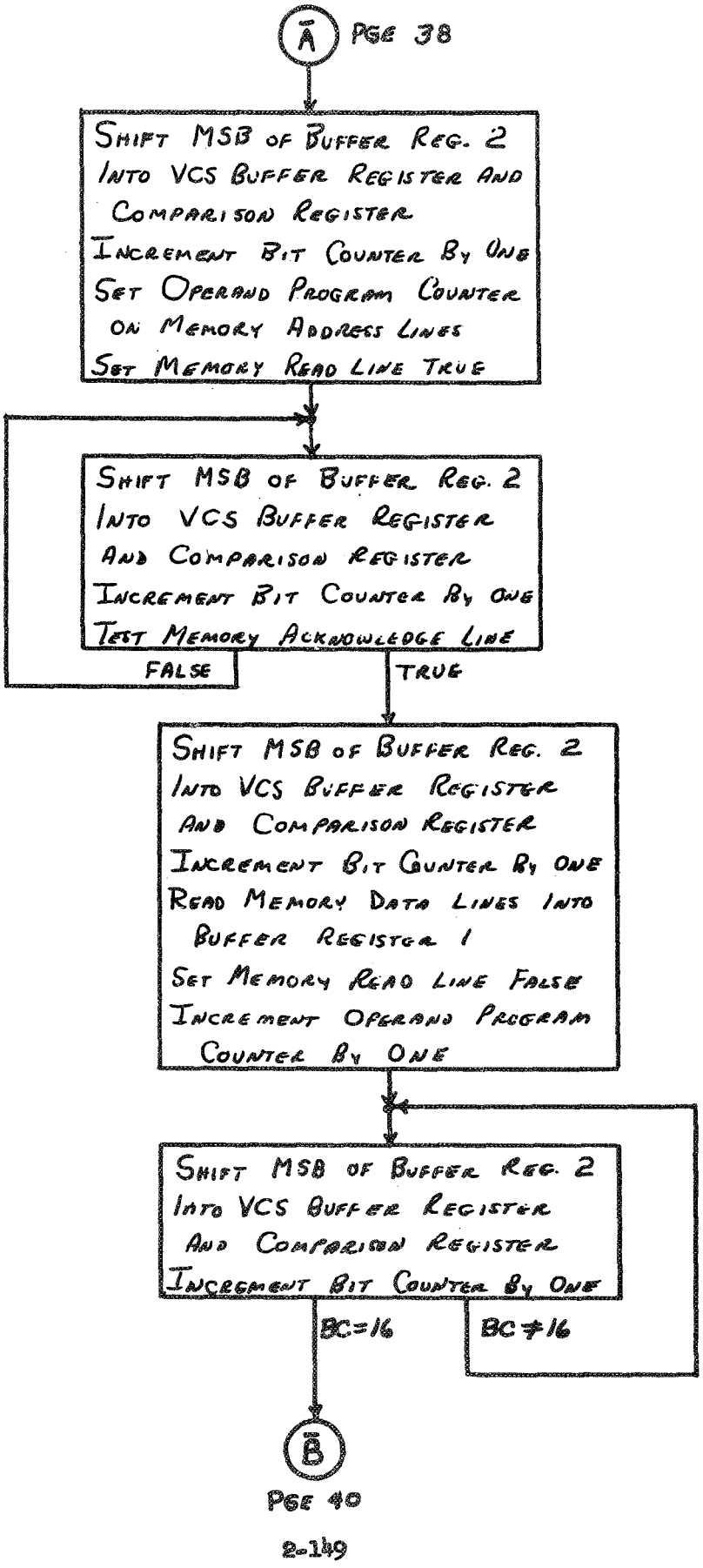

PGE 33
UU
SHIFT MSB OF BUFFER REG 2
ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
(BC=16)(WC=1)
TT
PGE 35
(BC=16)(WC>1)
GENERATE AND OUTPUT PARITY
ON TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
READ BUFFER REGISTER 1 INTO
BUFFER REGISTER 2
DECREMENT WORD COUNTER BY ONE
SS
PGE 33

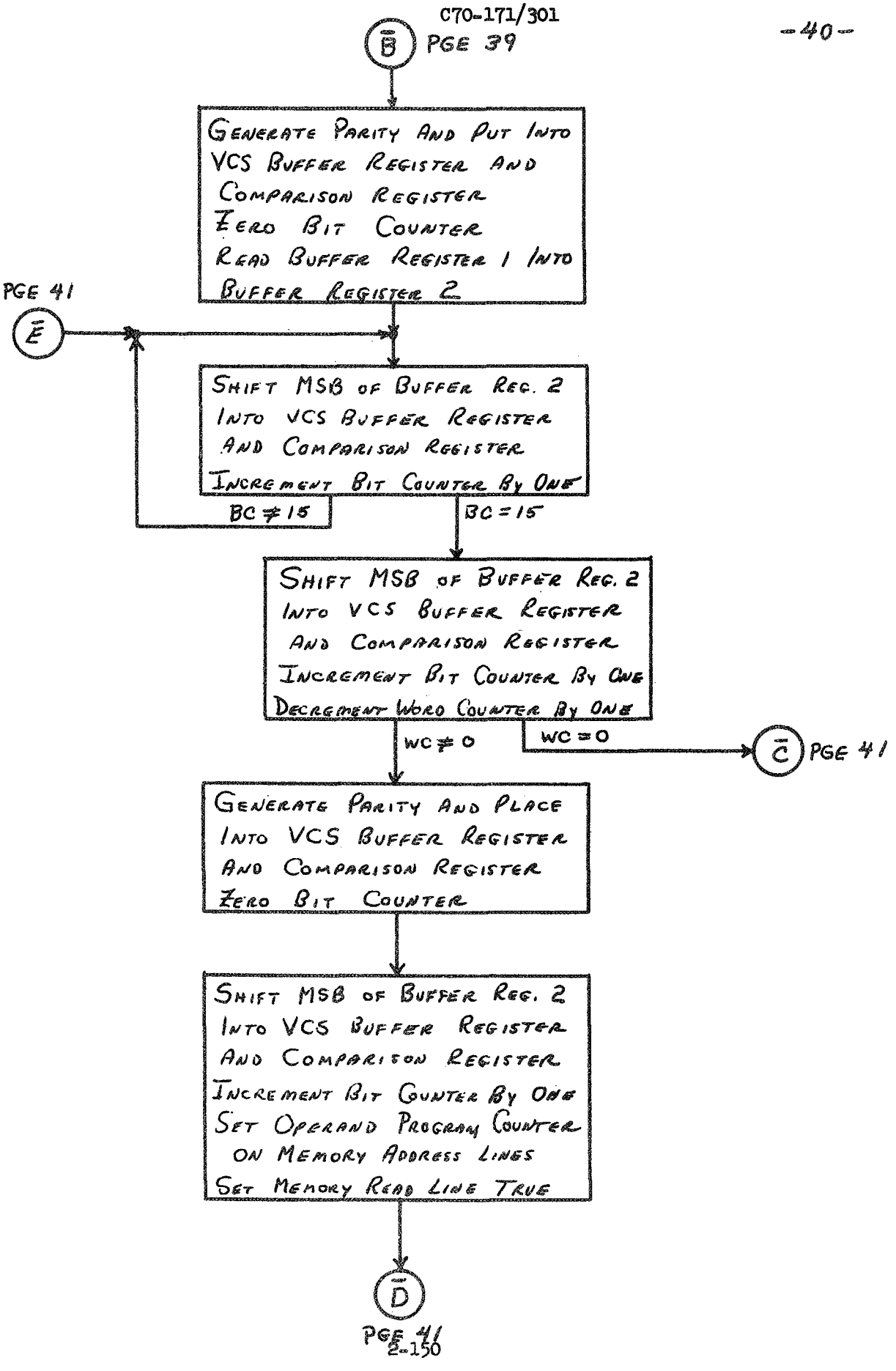

TT
PGE 33, 34
GENERATE AND OUTPUT PARITY ON TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
READ GO CODE INTO BUFFER REG-2
SHIFT MSB of BUFFER REG 2 ON TYPE 2 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY ON TYPE 2 OUTPUT LINE
ZERO BIT COUNTER
TEST FOR SIGNAL ON TYPE 2 INPUT LINE
SIGNAL
NO SIGNAL
INPUT ONE BIT OF SYNC CODE
INCREMENT BIT COUNTER BY ONE
BC ≠ 4
BC = 4
INPUT ONE BIT INTO BUFFER REG 2 FROM TYPE 2 INPUT LINE
INCREMENT BIT COUNTER BY ONE
BC = 37
BC ≠ 37
INPUT AND CHECK PARITY FROM TYPE 2 INPUT LINE
SET PARITY ERROR INDICATOR
ZERO BIT COUNTER
COMPARE BUFFER REG 2 WITH CONTAINED WORD REGISTER:
1. BITS 12 → N WITH BITS 21 → 25
2. " 21 → 26 " " 12 → 17
SET ACKNOWLEDGE WORD INDICATOR
SET ACKNOWLEDGE WORD ERROR INDICATOR
VV
PGE 36

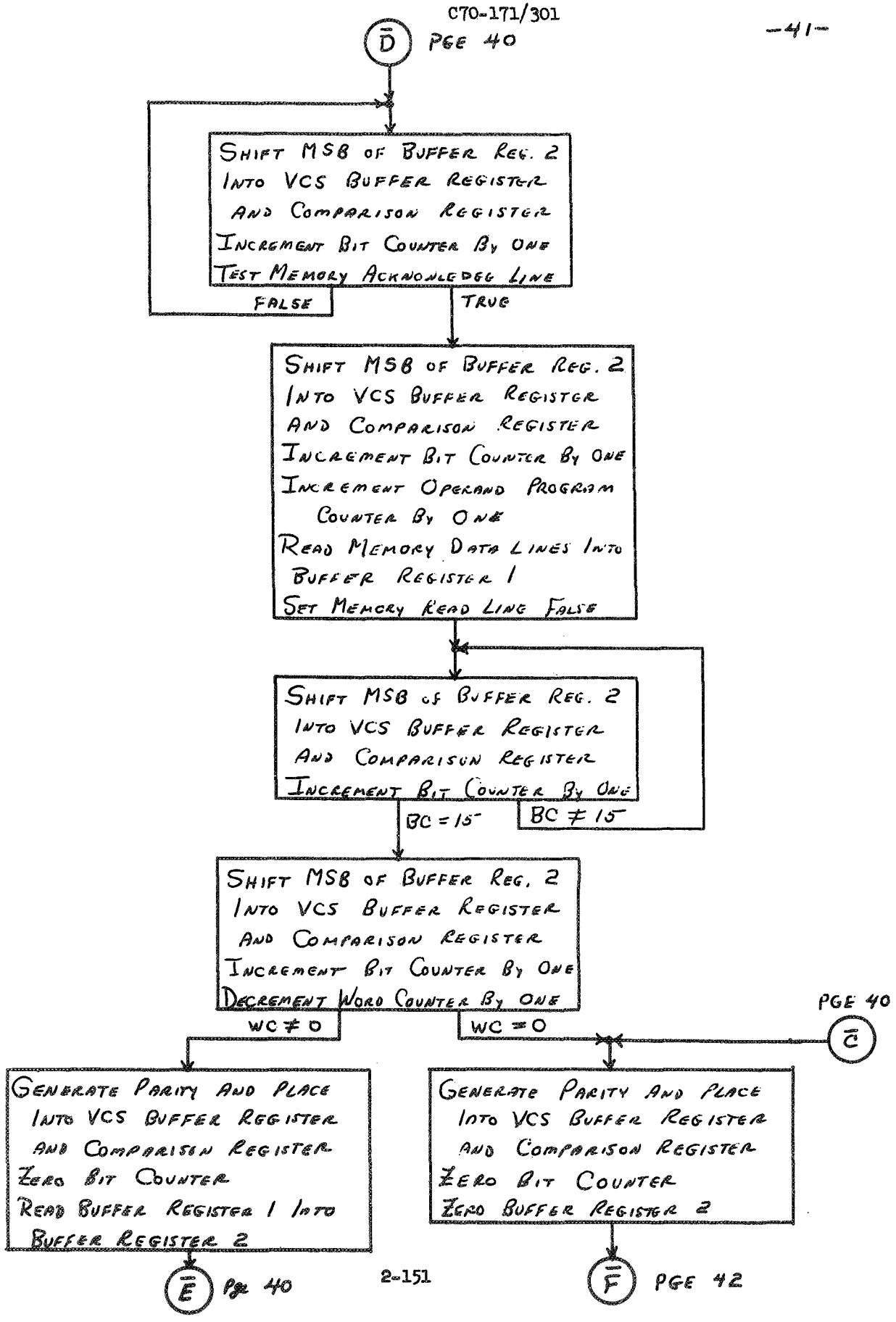

VV
PGE 35
FORM OUTPUT STATUS WORD IN BUFFER REGISTER 1
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
WW
PGE 31
SET MEMORY WRITE LINE FALSE
TEST BIT 26 OF CONTROL WORD REGISTER
ZERO
ONE
TEST ERROR INDICATORS
NONE SET
ANY SET
INCREMENT RETRANSMIT COUNTER BY ONE
SET RETRANSMIT INDICATOR
TEST RETRANSMIT COUNTER
RC=3
RC<3
INCREMENT COMMAND PROGRAM COUNTER BY ONE
IS A TYPE 1 INPUT INTERRUPT SET?
YES
NO
B
PGE 3
D
PGE 45

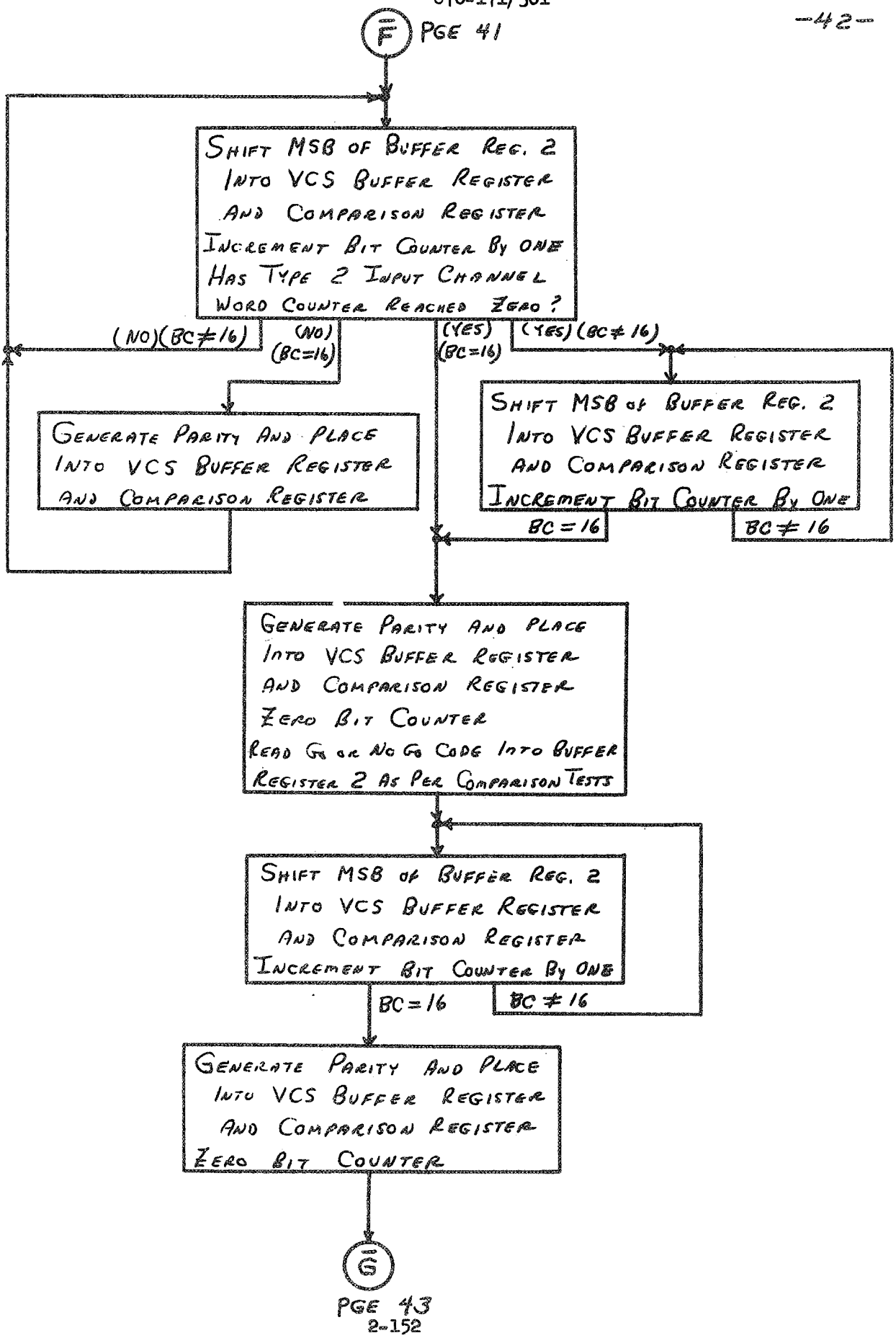

YY
PGE 11,47
TEST BITS 29 → 32 OF CONTROL WORD REGISTER
OTHER ADDRESS
OWN ADDRESS
ZZ
Pge 38
IS TYPE I OUTPUT CHANNEL BUSY ?
YES
NO
OUTPUT ONE BIT OF SYNC Code ON TYPE I OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
READ CONTROL WORD REGISTER INTO BUFFER REGISTER 2
OUTPUT ONE BIT OF Sync Code ON TYPE I OUTPUT LINE
INCREMENT BIT COUNTER by One
BC=4
BC≠4
SHIFT MSB of BUFFER REG 2 ON TYPE I OUTPUT LINE and INTO APPROPRIATE COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
TEST INPUT/OUTPUT FLIP FLOP
INPUT
OUTPUT
R
S
Pge 12
Pge 14

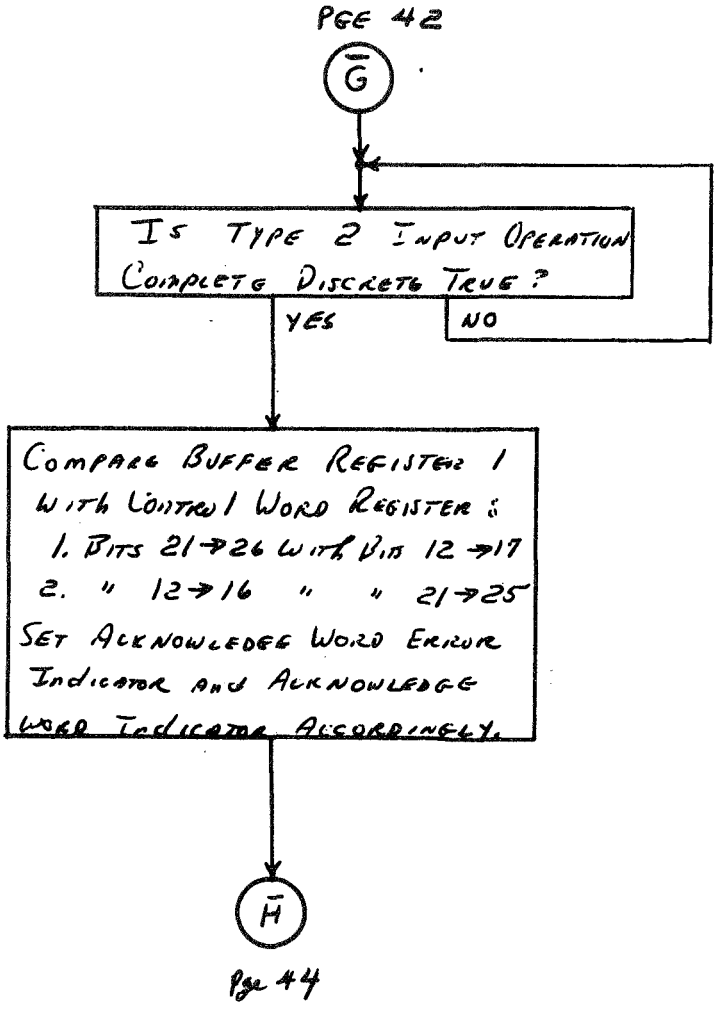

ZZ
PGE 37
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
SET OWN VCS INDICATOR
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
ZERO BIT COUNTER
TEST INPUT/OUTPUT INDICATOR
INPUT
OUTPUT
A
PGE 39
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
ZERO BIT COUNTER
IS TYPE 2 INPUT CHANNEL OPERATION COMPLETE DISCRETE SET?
IS COMPARISON INDICATOR SET?
EITHER SET
NEITHER SET
T
PGE 13

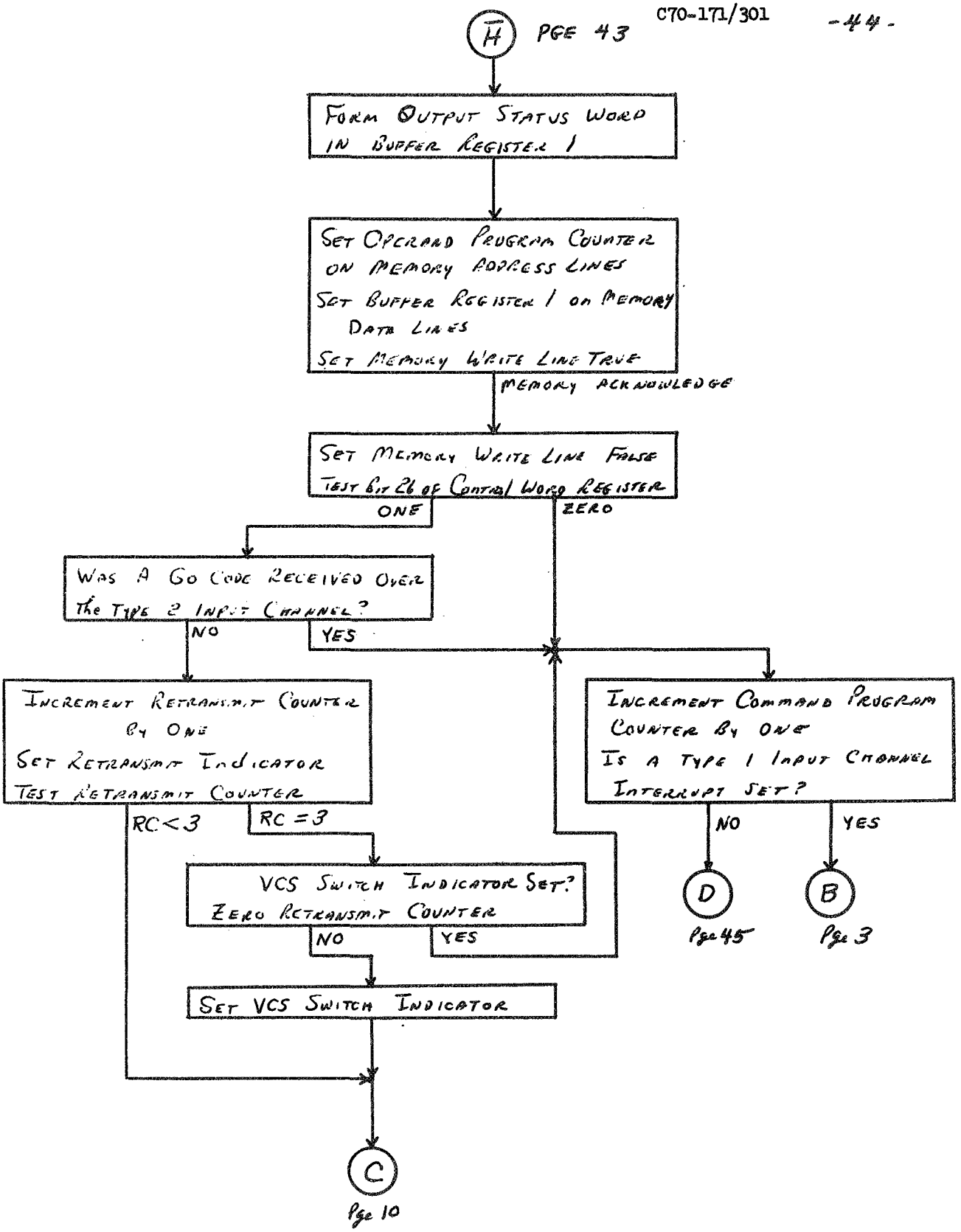

Ā
PGE 38
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE LINE
FALSE
TRUE
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC=16
BC≠16
B̄
PGE 40

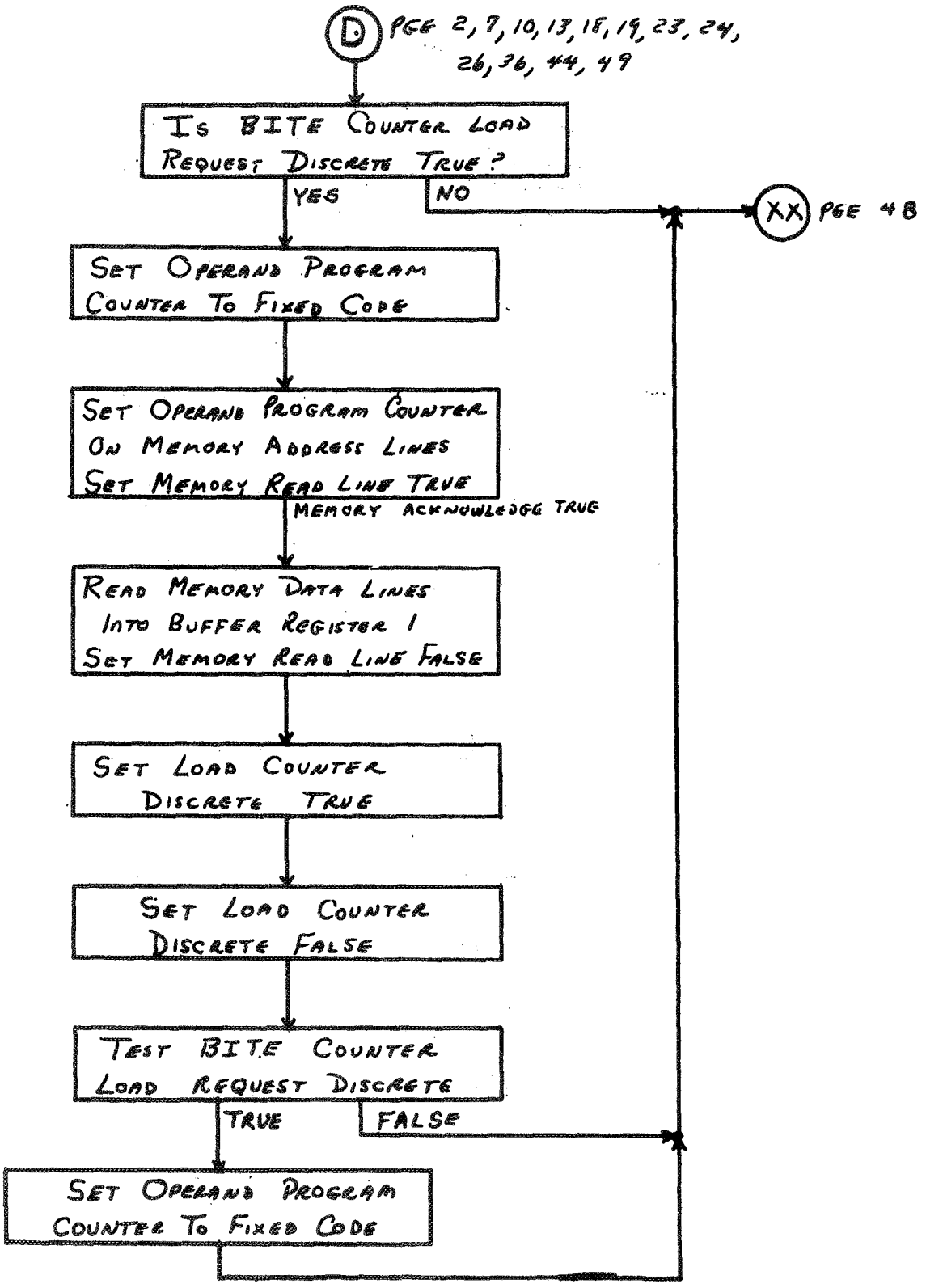

B
PGE 39
GENERATE PARITY AND PUT INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
ZERO BIT COUNTER
READ BUFFER REGISTER 1 INTO BUFFER REGISTER 2
PGE 41
E
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC ≠ 15
BC = 15
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
C
PGE 41
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
ZERO BIT COUNTER
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT GUNTER BY ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
D
PGE 41

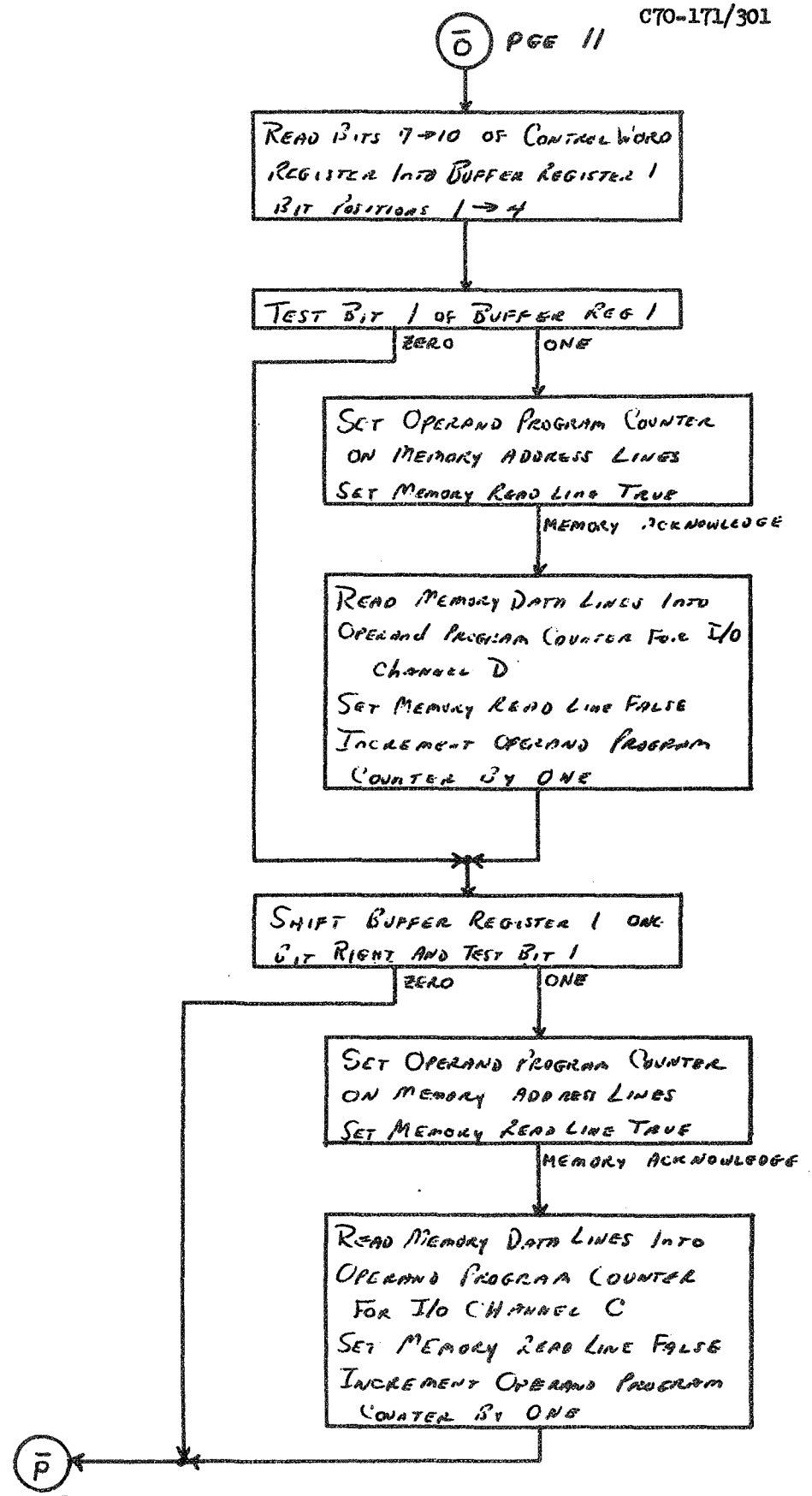

D
PGE 40
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
TEST MEMORY ACKNOWLEDGE LINE
FALSE
TRUE
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
BC = 15
BC ≠ 15
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
INCREMENT BIT COUNTER BY ONE
DECREMENT WORD COUNTER BY ONE
WC ≠ 0
WC = 0
PGE 40
C
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
ZERO BIT COUNTER
READ BUFFER REGISTER 1 INTO BUFFER REGISTER 2
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
ZERO BIT COUNTER
ZERO BUFFER REGISTER 2
E
Pge 40
F
PGE 42

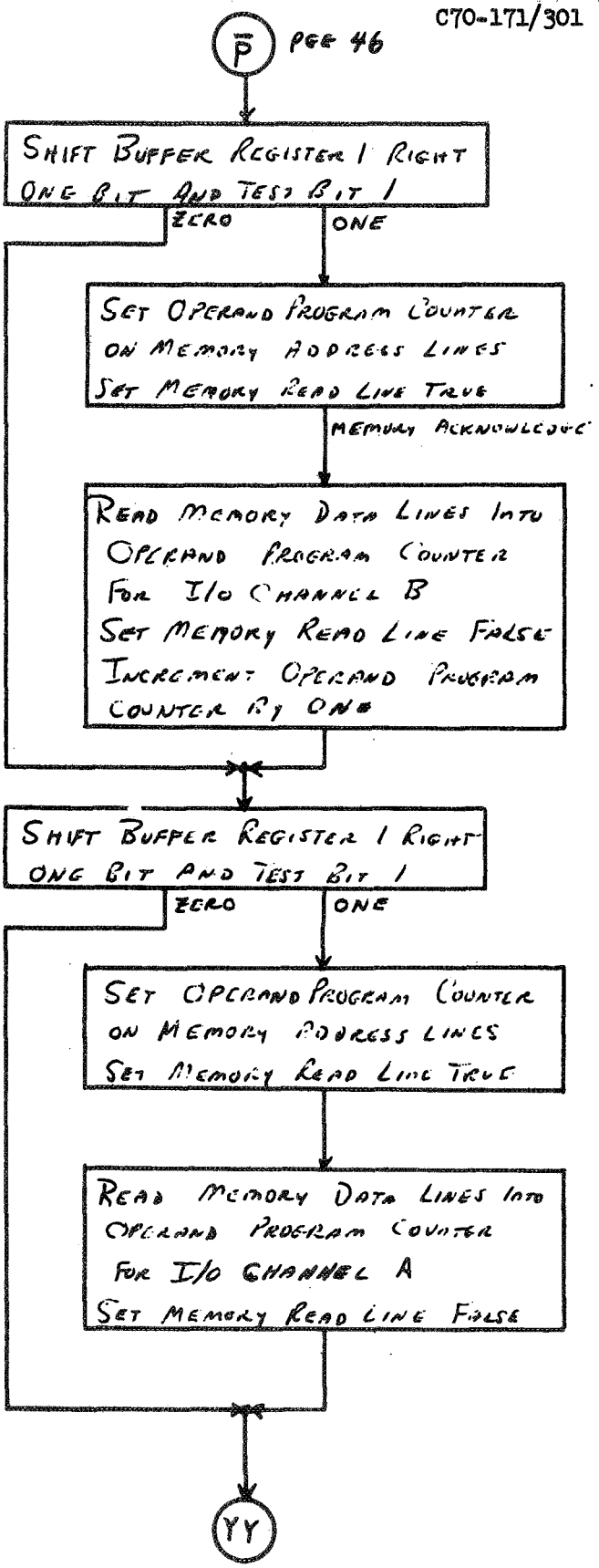

F
PGE 41
SHIFT MSB OF BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER INCREMENT BIT COUNTER BY ONE HAS TYPE 2 INPUT CHANNEL WORD COUNTER REACHED ZERO ?
(NO)(BC≠16)
(NO) (BC=16)
(YES) (BC=16)
(YES)(BC≠16)
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER
SHIFT MSB of BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER ZERO BIT COUNTER READ Go or No Go CODE INTO BUFFER REGISTER 2 AS PER COMPARISON TESTS
SHIFT MSB of BUFFER REG. 2 INTO VCS BUFFER REGISTER AND COMPARISON REGISTER INCREMENT BIT COUNTER BY ONE
BC = 16
BC ≠ 16
GENERATE PARITY AND PLACE INTO VCS BUFFER REGISTER AND COMPARISON REGISTER ZERO BIT COUNTER
G
PGE 43

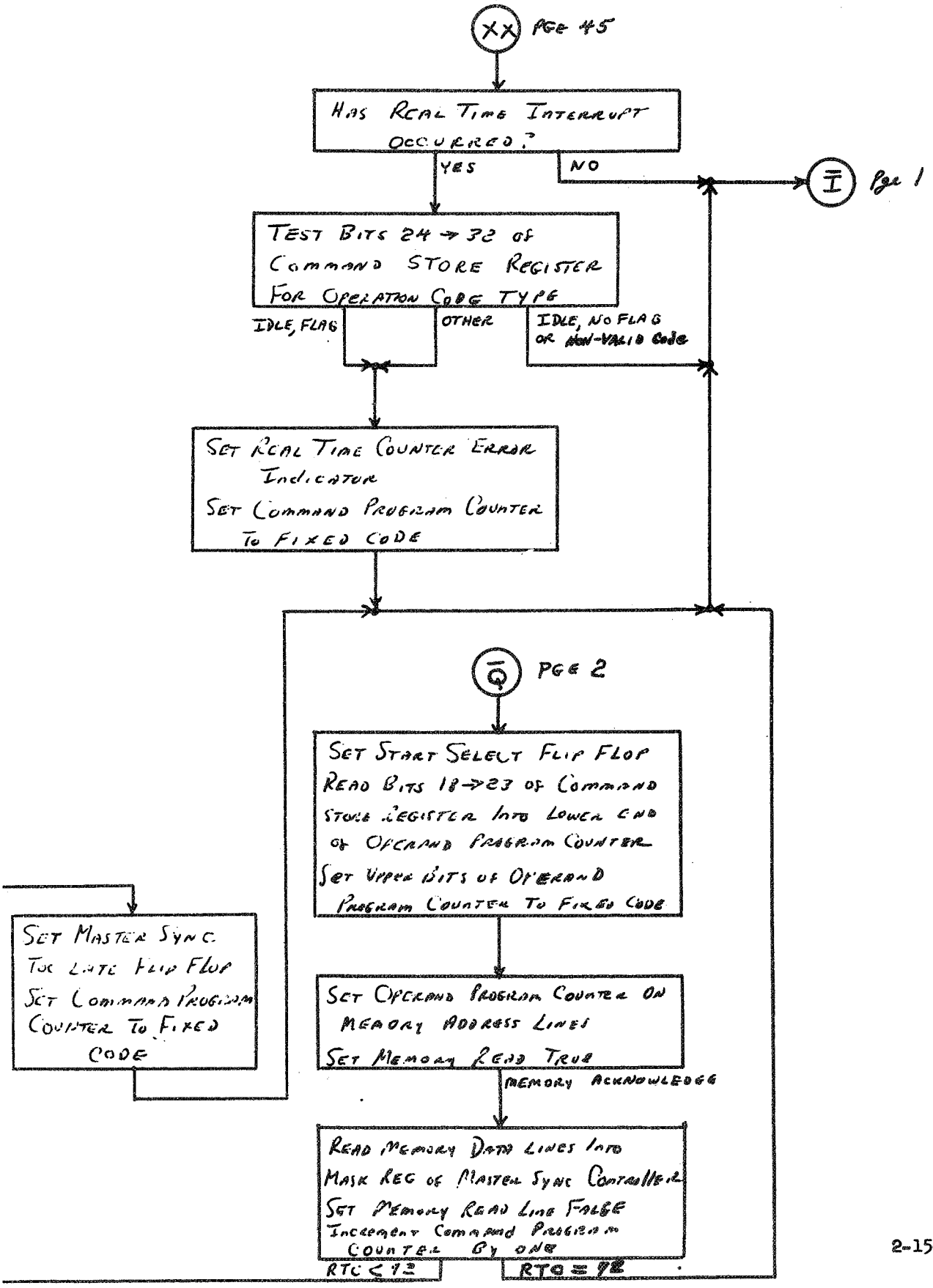

PGE 42
G
IS TYPE 2 INPUT OPERATION COMPLETE DISCRETE TRUE ?
YES
NO
COMPARE BUFFER REGISTER 1 WITH CONTROL WORD REGISTER :
1. BITS 21→26 WITH BITS 12→17
2. " 12→16 " " 21→25
SET ACKNOWLEDGE WORD ERROR INDICATOR AND ACKNOWLEDGE WORD INDICATOR ACCORDINGLY.
H
Pge 44

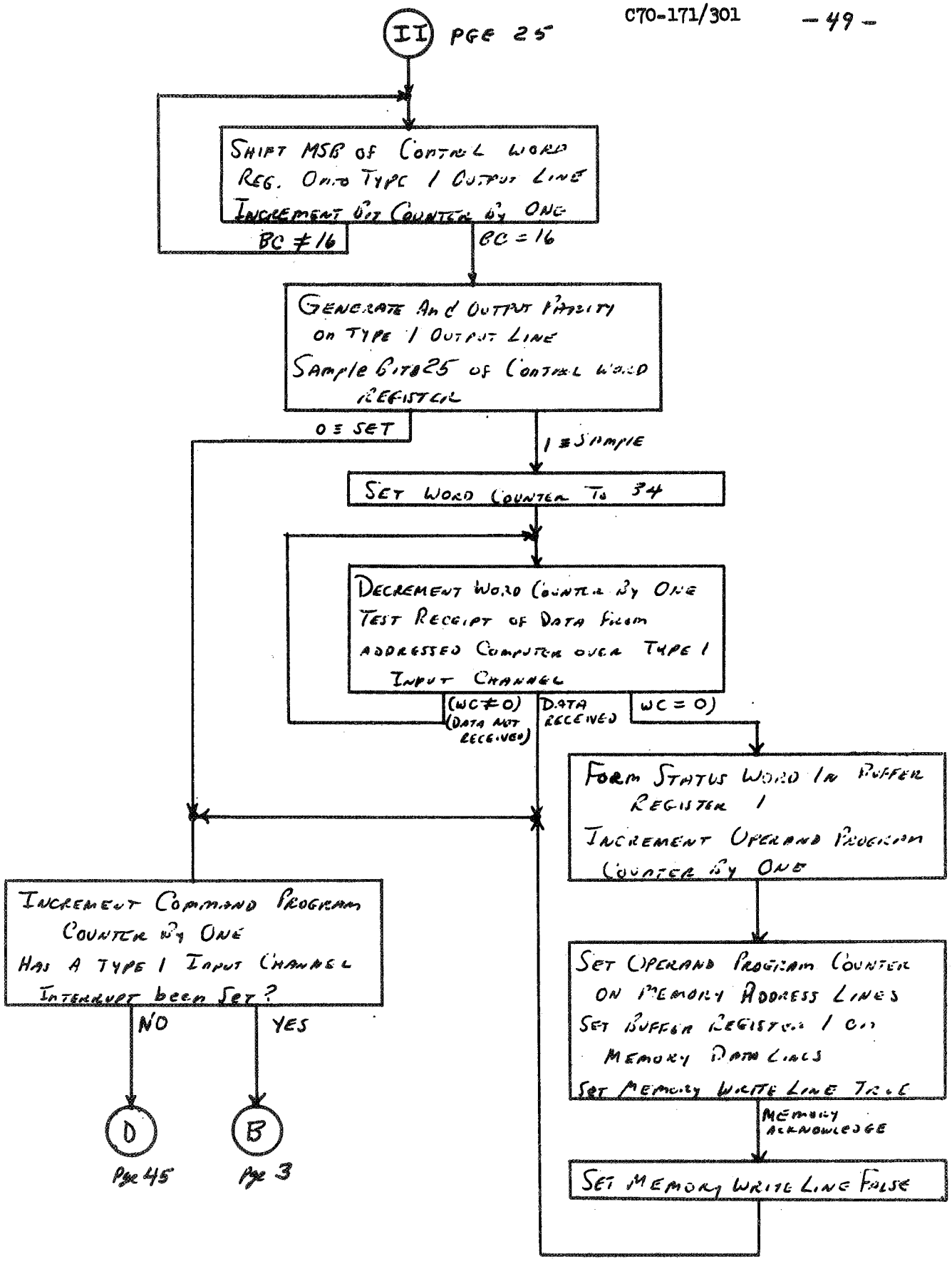

H
PGE 43
FORM OUTPUT STATUS WORD IN BUFFER REGISTER 1
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
SET MEMORY WRITE LINE FALSE
TEST BIT 26 OF CONTROL WORD REGISTER
ONE
ZERO
WAS A GO CODE RECEIVED OVER THE TYPE 2 INPUT CHANNEL?
NO
YES
INCREMENT RETRANSMIT COUNTER BY ONE
SET RETRANSMIT INDICATOR
TEST RETRANSMIT COUNTER
RC<3
RC=3
VCS SWITCH INDICATOR SET? ZERO RETRANSMIT COUNTER
NO
YES
SET VCS SWITCH INDICATOR
INCREMENT COMMAND PROGRAM COUNTER BY ONE
IS A TYPE 1 INPUT CHANNEL INTERRUPT SET?
NO
YES
D
Pge 45
B
Pge 3
C
Pge 10

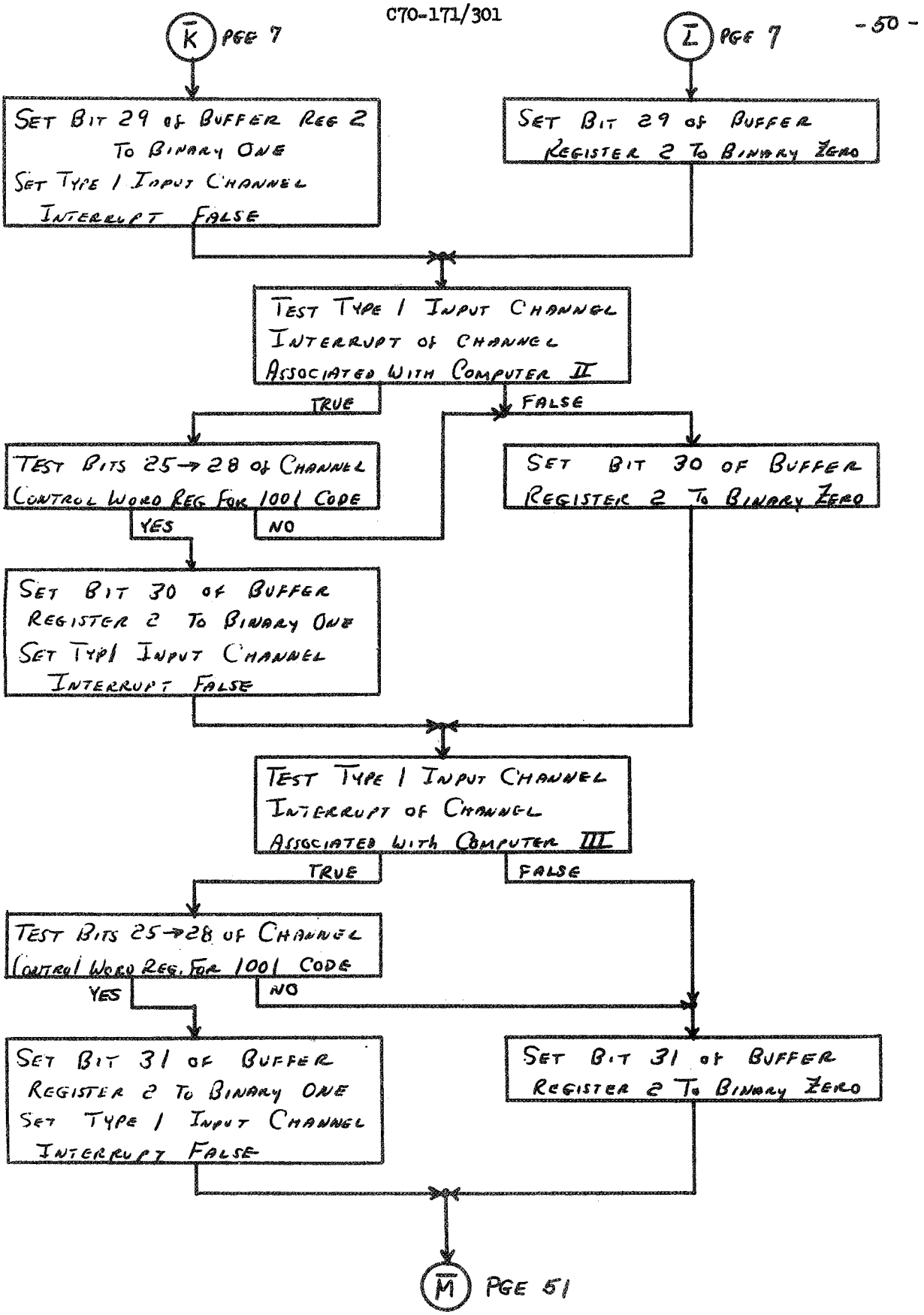

D
PGE 2, 7, 10, 13, 18, 19, 23, 24, 26, 36, 44, 49
IS BITE COUNTER LOAD REQUEST DISCRETE TRUE?
YES
NO
XX
PGE 48
SET OPERAND PROGRAM COUNTER TO FIXED CODE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE TRUE
READ MEMORY DATA LINES INTO BUFFER REGISTER 1
SET MEMORY READ LINE FALSE
SET LOAD COUNTER DISCRETE TRUE
SET LOAD COUNTER DISCRETE FALSE
TEST BITE COUNTER LOAD REQUEST DISCRETE
TRUE
FALSE
SET OPERAND PROGRAM COUNTER TO FIXED CODE

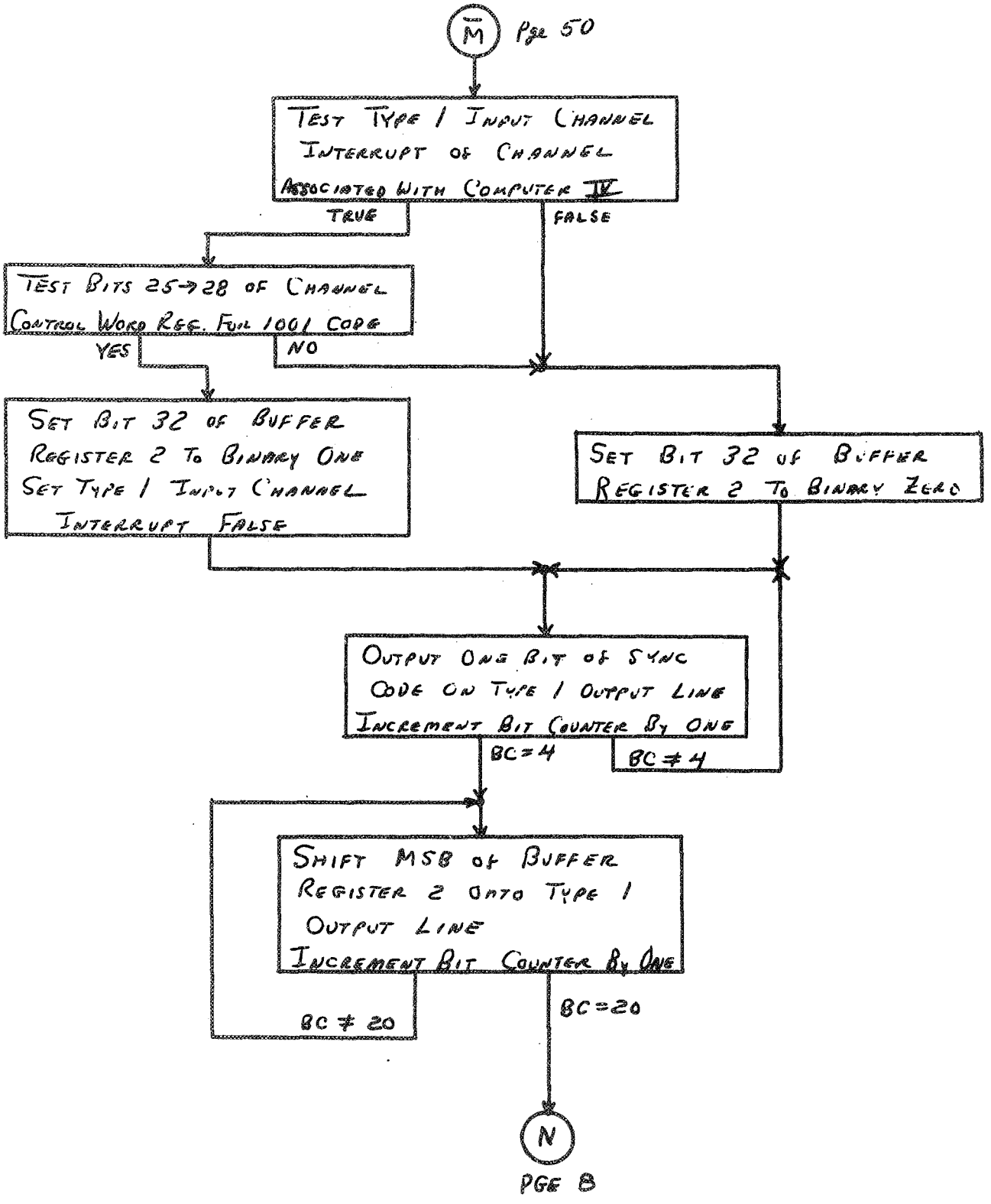

O
PGE 11
READ BITS 7→10 OF CONTROL WORD REGISTER INTO BUFFER REGISTER 1 BIT POSITIONS 1→4
TEST BIT 1 OF BUFFER REG 1
ZERO
ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO OPERAND PROGRAM COUNTER FOR I/O CHANNEL D
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SHIFT BUFFER REGISTER 1 ONE BIT RIGHT AND TEST BIT 1
ZERO
ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO OPERAND PROGRAM COUNTER FOR I/O CHANNEL C
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
P
Pge 47


P
PGE 46
SHIFT BUFFER REGISTER 1 RIGHT ONE BIT AND TEST BIT 1
ZERO
ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO OPERAND PROGRAM COUNTER FOR I/O CHANNEL B
SET MEMORY READ LINE FALSE
INCREMENT OPERAND PROGRAM COUNTER BY ONE
SHIFT BUFFER REGISTER 1 RIGHT ONE BIT AND TEST BIT 1
ZERO
ONE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ LINE TRUE
READ MEMORY DATA LINES INTO OPERAND PROGRAM COUNTER FOR I/O CHANNEL A
SET MEMORY READ LINE FALSE
YY
Pg 37

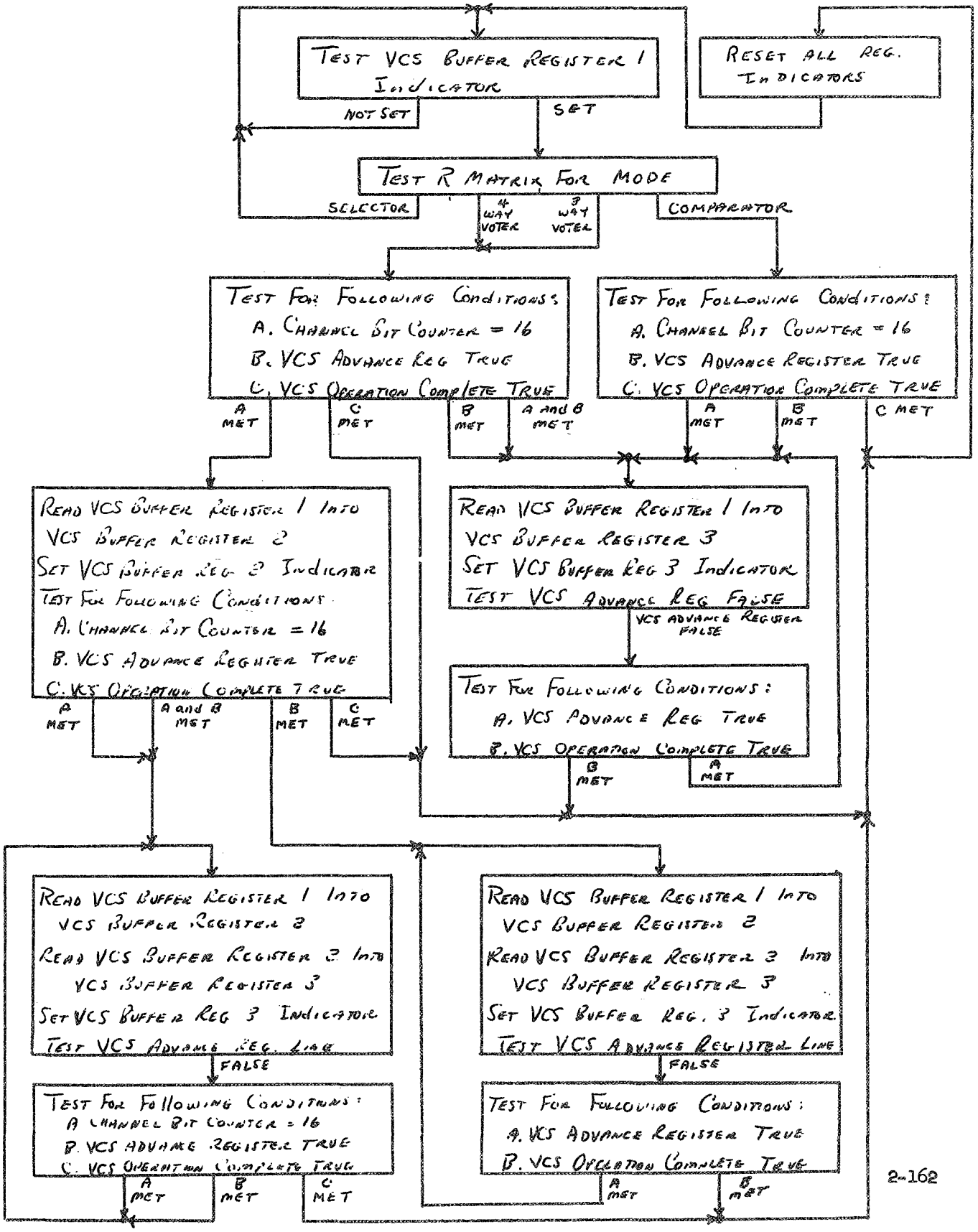

XX
PGE 45
HAS REAL TIME INTERRUPT OCCURRED?
YES
NO
I
Pge 1
TEST BITS 24 → 32 OF COMMAND STORE REGISTER FOR OPERATION CODE TYPE
IDLE, FLAG
OTHER
IDLE, NO FLAG OR NON-VALID CODE
SET REAL TIME COUNTER ERROR INDICATOR
SET COMMAND PROGRAM COUNTER TO FIXED CODE
Q
PGE 2
SET START SELECT FLIP FLOP
READ BITS 18 → 23 OF COMMAND STORE REGISTER INTO LOWER END OF OPERAND PROGRAM COUNTER
SET UPPER BITS OF OPERAND PROGRAM COUNTER TO FIXED CODE
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET MEMORY READ TRUE
MEMORY ACKNOWLEDGE
READ MEMORY DATA LINES INTO MASK REG OF MASTER SYNC CONTROLLER
SET MEMORY READ LINE FALSE
INCREMENT COMMAND PROGRAM COUNTER BY ONE
RTC < 72
RTC = 72
SET MASTER SYNC TOC LATE FLIP FLOP
SET COMMAND PROGRAM COUNTER TO FIXED CODE


II PGE 25
SHIFT MSB OF CONTROL WORD REG. ONTO TYPE 1 OUTPUT LINE
INCREMENT BIT COUNTER BY ONE
BC ≠ 16
BC = 16
GENERATE AND OUTPUT PARITY ON TYPE 1 OUTPUT LINE
SAMPLE BIT 25 OF CONTROL WORD REGISTER
0 = SET
1 = SAMPLE
SET WORD COUNTER TO 34
DECREMENT WORD COUNTER BY ONE
TEST RECEIPT OF DATA FROM ADDRESSED COMPUTER OVER TYPE 1 INPUT CHANNEL
(WC ≠ 0) (DATA NOT RECEIVED)
DATA RECEIVED
WC = 0)
FORM STATUS WORD IN BUFFER REGISTER 1
INCREMENT OPERAND PROGRAM COUNTER BY ONE
INCREMENT COMMAND PROGRAM COUNTER BY ONE
HAS A TYPE 1 INPUT CHANNEL INTERRUPT been SET?
NO
YES
SET OPERAND PROGRAM COUNTER ON MEMORY ADDRESS LINES
SET BUFFER REGISTER 1 ON MEMORY DATA LINES
SET MEMORY WRITE LINE TRUE
MEMORY ACKNOWLEDGE
SET MEMORY WRITE LINE FALSE
D
Pge 45
B
Pge 3

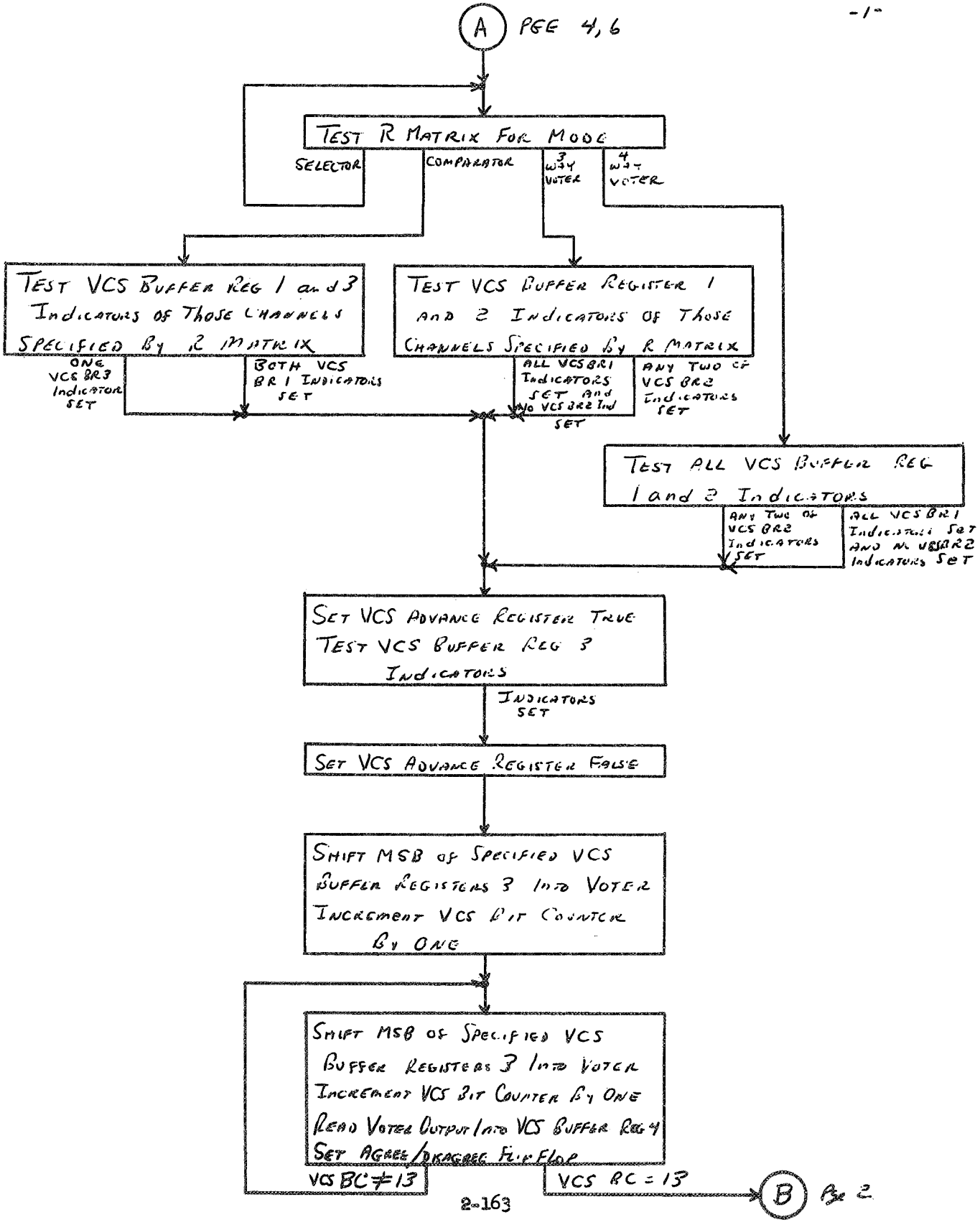

K PGE 7
L PGE 7
SET BIT 29 of BUFFER REG 2 TO BINARY ONE
SET TYPE 1 INPUT CHANNEL INTERRUPT FALSE
SET BIT 29 of BUFFER REGISTER 2 TO BINARY ZERO
TEST TYPE 1 INPUT CHANNEL INTERRUPT of CHANNEL ASSOCIATED WITH COMPUTER II
TRUE
FALSE
TEST BITS 25→28 of CHANNEL CONTROL WORD REG FOR 1001 CODE
SET BIT 30 OF BUFFER REGISTER 2 TO BINARY ZERO
YES
NO
SET BIT 30 of BUFFER REGISTER 2 TO BINARY ONE
SET TYP1 INPUT CHANNEL INTERRUPT FALSE
TEST TYPE 1 INPUT CHANNEL INTERRUPT of CHANNEL ASSOCIATED WITH COMPUTER III
TRUE
FALSE
TEST BITS 25→28 of CHANNEL CONTROL WORD REG. FOR 1001 CODE
YES
NO
SET BIT 31 of BUFFER REGISTER 2 TO BINARY ONE
SET TYPE 1 INPUT CHANNEL INTERRUPT FALSE
SET BIT 31 of BUFFER REGISTER 2 TO BINARY ZERO
M PGE 51

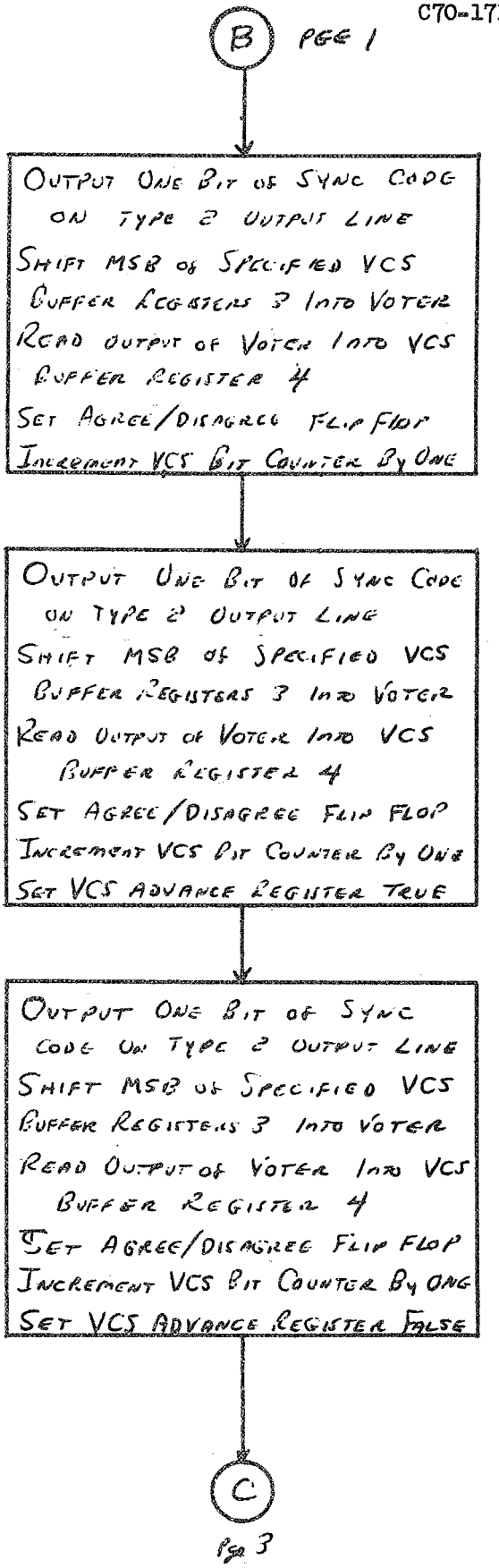

M
Pge 50
TEST TYPE 1 INPUT CHANNEL INTERRUPT OF CHANNEL ASSOCIATED WITH COMPUTER IV
TRUE
FALSE
TEST BITS 25→28 OF CHANNEL CONTROL WORD REG. FOR 1001 CODE
YES
NO
SET BIT 32 OF BUFFER REGISTER 2 TO BINARY ONE SET TYPE 1 INPUT CHANNEL INTERRUPT FALSE
SET BIT 32 OF BUFFER REGISTER 2 TO BINARY ZERO
OUTPUT ONE BIT OF SYNC CODE ON TYPE 1 OUTPUT LINE INCREMENT BIT COUNTER BY ONE
BC=4
BC≠4
SHIFT MSB OF BUFFER REGISTER 2 ONTO TYPE 1 OUTPUT LINE INCREMENT BIT COUNTER BY ONE
BC≠20
BC=20
N
PGE 8

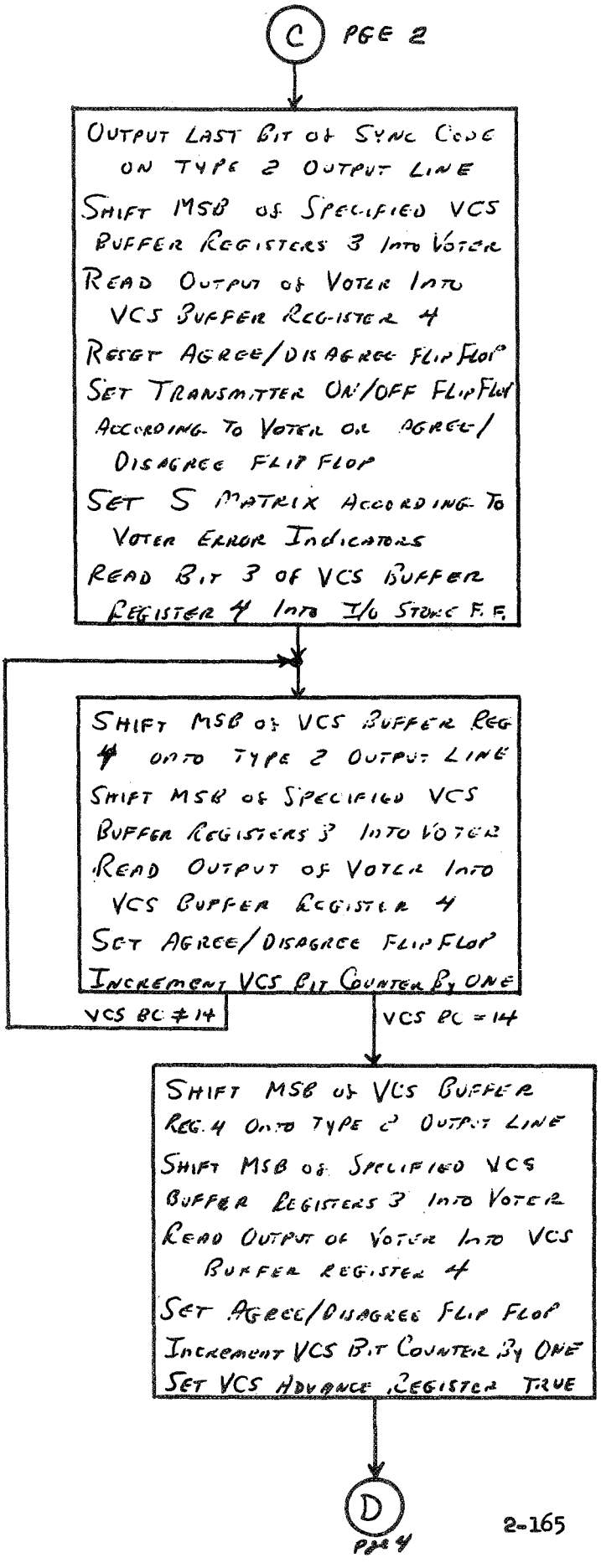
4.2.7 VCS Input Channel

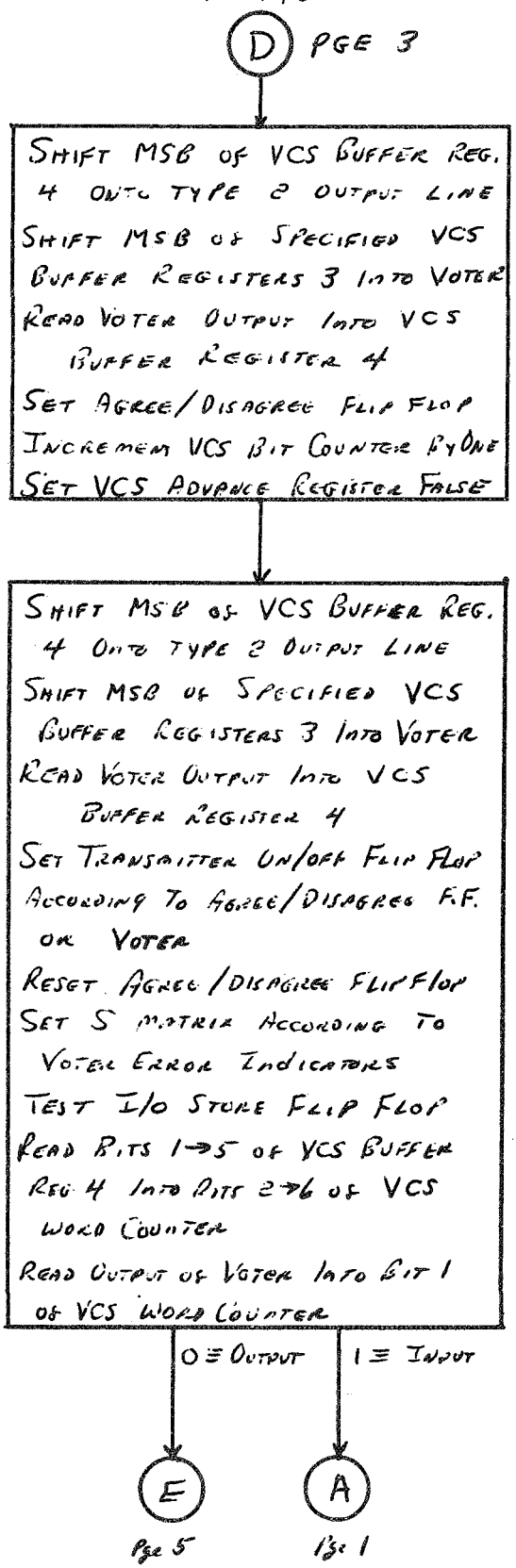


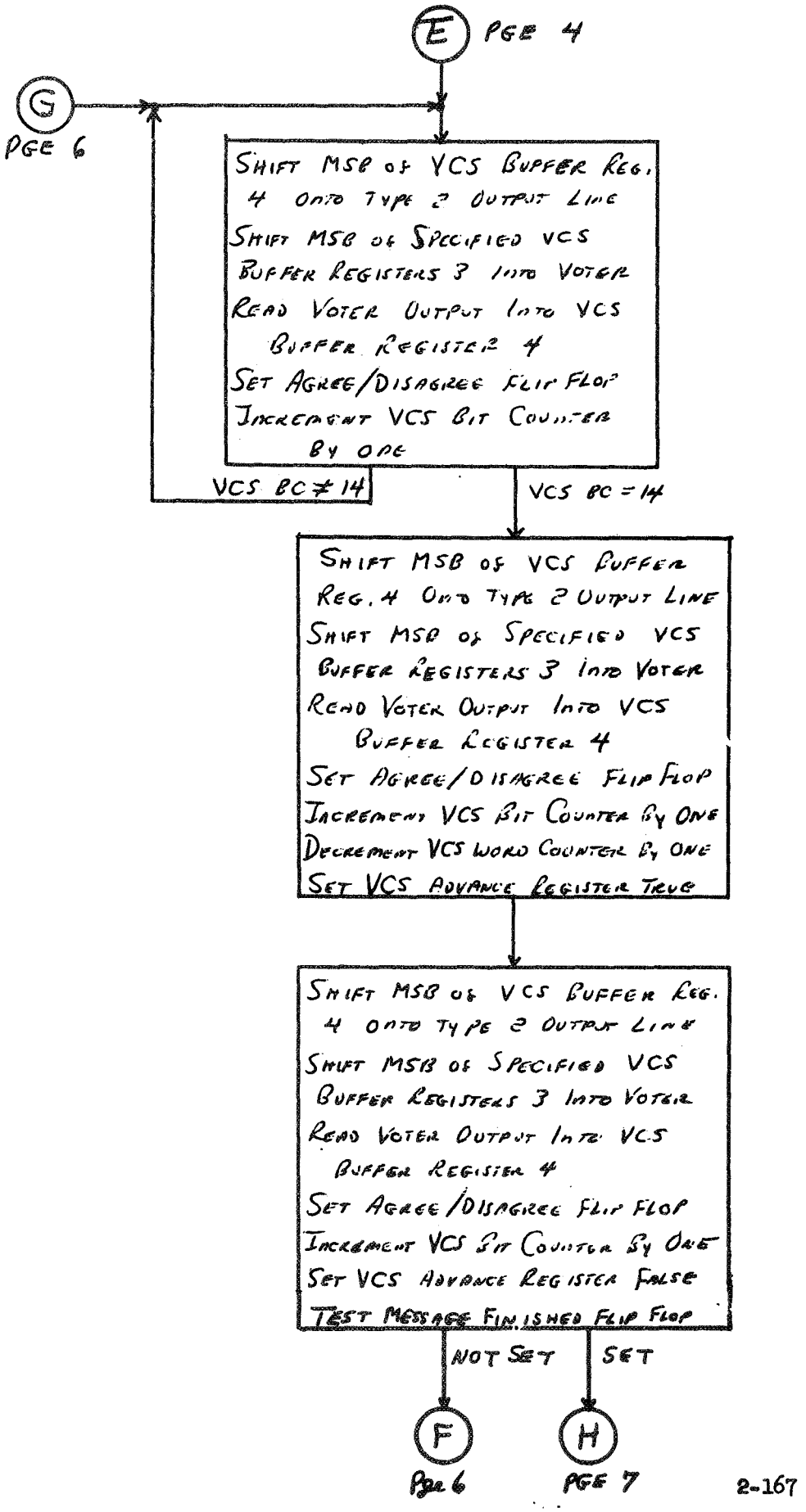
4.2.8 VCS Voter and Output Channel

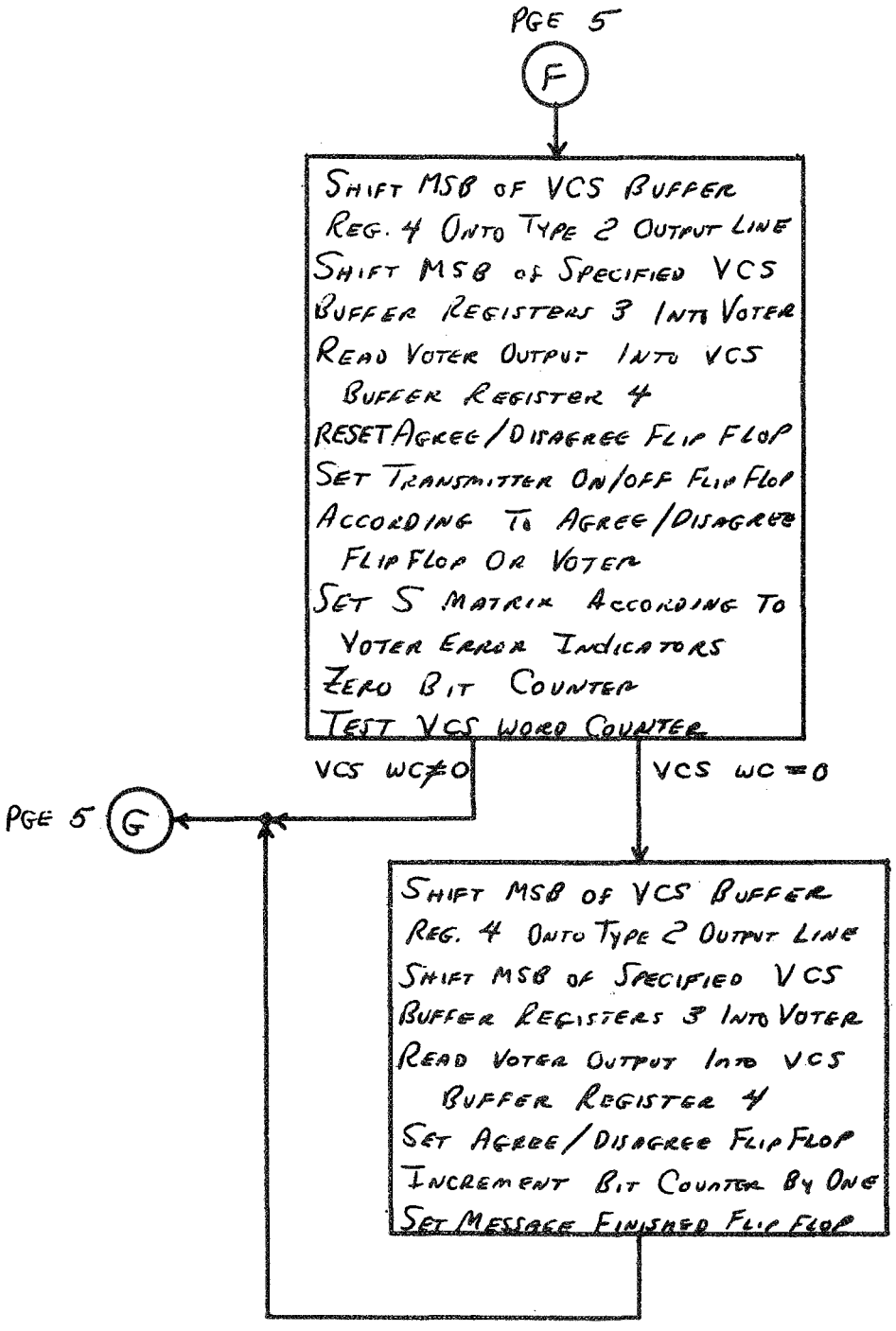



B
PGE 1
OUTPUT ONE BIT OF SYNC CODE ON TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ OUTPUT OF VOTER INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
OUTPUT ONE BIT OF SYNC CODE ON TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ OUTPUT OF VOTER INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
SET VCS ADVANCE REGISTER TRUE
OUTPUT ONE BIT OF SYNC CODE ON TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ OUTPUT OF VOTER INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
SET VCS ADVANCE REGISTER FALSE
C
Pge 3


C
PGE 2
OUTPUT LAST BIT OF SYNC CODE ON TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ OUTPUT OF VOTER INTO VCS BUFFER REGISTER 4
RESET AGREE/DISAGREE FLIP FLOP
SET TRANSMITTER ON/OFF FLIP FLOP ACCORDING TO VOTER OR AGREE/DISAGREE FLIP FLOP
SET S MATRIX ACCORDING TO VOTER ERROR INDICATORS
READ BIT 3 OF VCS BUFFER REGISTER 4 INTO I/O STORE F.F.
SHIFT MSB OF VCS BUFFER REG 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ OUTPUT OF VOTER INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
VCS BC ≠ 14
VCS BC = 14
SHIFT MSB OF VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ OUTPUT OF VOTER INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
SET VCS ADVANCE REGISTER TRUE
D
PGE 4


D
PGE 3
SHIFT MSB OF VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
SET VCS ADVANCE REGISTER FALSE
SHIFT MSB OF VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB OF SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
SET TRANSMITTER ON/OFF FLIP FLOP ACCORDING TO AGREE/DISAGREE F.F. OR VOTER
RESET AGREE/DISAGREE FLIP FLOP
SET S MATRIX ACCORDING TO VOTER ERROR INDICATORS
TEST I/O STORE FLIP FLOP
READ BITS 1→5 OF VCS BUFFER REG 4 INTO BITS 2→6 OF VCS WORD COUNTER
READ OUTPUT OF VOTER INTO BIT 1 OF VCS WORD COUNTER
0 ≡ OUTPUT
1 ≡ INPUT
E
Pge 5
A
Pge 1


E
PGE 4
G
PGE 6
SHIFT MSB of VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB of SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
VCS BC ≠ 14
VCS BC = 14
SHIFT MSB of VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB of SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
DECREMENT VCS WORD COUNTER BY ONE
SET VCS ADVANCE REGISTER TRUE
SHIFT MSB of VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB of SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT VCS BIT COUNTER BY ONE
SET VCS ADVANCE REGISTER FALSE
TEST MESSAGE FINISHED FLIP FLOP
NOT SET
SET
F
Pge 6
H
PGE 7


PGE 5
F
SHIFT MSB OF VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB of SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
RESET AGREE/DISAGREE FLIP FLOP
SET TRANSMITTER ON/OFF FLIP FLOP ACCORDING TO AGREE/DISAGREE FLIP FLOP OR VOTER
SET S MATRIX ACCORDING TO VOTER ERROR INDICATORS
ZERO BIT COUNTER
TEST VCS WORD COUNTER
VCS WC≠0
VCS WC=0
PGE 5
G
SHIFT MSB OF VCS BUFFER REG. 4 ONTO TYPE 2 OUTPUT LINE
SHIFT MSB of SPECIFIED VCS BUFFER REGISTERS 3 INTO VOTER
READ VOTER OUTPUT INTO VCS BUFFER REGISTER 4
SET AGREE/DISAGREE FLIP FLOP
INCREMENT BIT COUNTER BY ONE
SET MESSAGE FINISHED FLIP FLOP

PGE 5